

PTK-1096



KEZELÉSI ÚTMUTATÓ

BILLENTYÖZET

A'	B'	C'	D'	E'
A	B	C	D	E
2nd	INV	log lnx	CP CE	CLR
Pgm LRN	P→R x↔t	sin x ²	cos √x	tan 1/x
Ins SST	CMs STO	Exc RCL	Prd SUM	Ind y ^x
Del BST	Eng EE	Fix (	Int)	IxI ÷
Pause GTO	x=t 7	Nop 8	Op 9	Deg X
Lbl SBR	x≥t 4	Σ+ 5	$\bar{x}$ 6	Rad -
St flg RST	If flg 1	D.MS 2	π 3	Grad +
Write RIS	Dsz 0	Adv •	Prt +/-	List =

TARTALOMJEGYZÉK

5	I. Alapismeretek
	Bevezetés
	Kezelési utasítás
6	Üzem módok
	A software egységek felhasználása
7	Számolás a billentyűzet segítségével
	Példa saját program készítésére
	Nyomtatási lehetőségek
8	Számológépes rendszer felépítése
	II. Billentyűzet
	A billentyűzet fontosabb tulajdonságai
	A kijelzés törlése CE, CLR
	Adatbeíró billentyűk 0..9, ., +/-, π
	Az alpműveletek billentyűi +, -, X, $\div$, =
9	Az AOS beírási módszer
	A zárójel-billentyűk (;)
10	A többszörös rendeltetésű billentyűk vezérlése 2nd INV
	Tárolóbillentyűk CMs, STO, RCL, Exc
11	Műveletek az adattárolókban SUM, Prd
	A kijelzés vezérlése
	Alap kijelzési mód
12	Exponenciális kijelzési mód EE
	Műszaki kijelzés Eng
13	Meghatározott számú tizedesjegy beállítása Fix
	Matematikai függvények
	Négyzet, négyzetgyökök és reciprok érték x^2 , $\sqrt{x}$, $1/x$
	Hatványok és gyökök
14	A hatványozás lépései y^x
	A gyökvonás lépései INV, y^x
	Logaritmusok ln x, log
	Szögérték-billentyűk Deg, Rad, Grad
15	A trigonometrikus függvények billentyűi sin, cos, tan
	Átszámítások
	Régi fok átszámítása D.Ms
	Poláris és derékszögű koordináták átszámítása P $\rightarrow$ R
17	Statistikai műveletek billentyűi
	Középérték, szórásnégyzet, szórás $\bar{x}$, σ_p , σ_s , σ_{11} , $\sigma_{11} \sqrt{x}$
	Lineáris regresszió σ_p 12..15
18	III. A „beépített” feladatmegoldások alkalmazása
	Bevezetés a szilárdtest-áramkörű. software programokba
	A software-egység
19	A software-programok lefuttatása
	A software-programok elemzése (programátvétel)
20	IV. Programozás
	A programozás fogalma

21	A programozás alapjai
	Változók bevétele a programba
28	A programozás gyakorlata
	A programcím-billentyűk alkalmazása
31	Címzés rövidített formában
	Egyéni program billentyűzése
32	A program kijelzése
33	Időkülönbség-program
35	A programok átdolgozása
	Az időkülönbség-program javítása
38	Programátírás kombinált kódokkal
39	Jellegzetes programozási példák
42	Árszámítási program
45	Program gömbkoordináták átszámítására
48	Haladó programozási feladatok
	További részletek a címkékről
49	Elágazási utasítás
	Feltétel nélküli elágazások
	A „go-to” utasítás GTO
50	Alprogramok
	Az alprogramutasítás SBR
51	Az alprogramok behívása
53	Software-programok, mint alprogramok
54	Bioritmus-program
57	Feltételhez kötött elágazások
	A kijelző-tároló és a t-tároló összehasonlítása
58	Négyzetgyökvonás
60	A flag (jelző) művelet
62	A jelzők különleges rendeltetés
63	Metrikus egységek átszámítása
64	Adattároló-elágazások Dsz
65	Programhurok készítése
	Feltétel nélküli hurokképzés
	Feltételhez kötött hurokképzés
66	Hurokképzés a feltételhez kötött elágaztatási utasítással Dsz
67	Az xI program
69	A másodfokú egyenletek megoldásának programja
73	További programozási módszerek
74	Közvetett utasítások programozása
	Közvetett művelet az adattárolóval
74	Közvetett elágazási utasítások
75	Egyéb tulajdonságok
	A program optimalizálása
	Programozási technikák egyszerűsített alkalmazása
	Programozási módszerek a lépésszám csökkentésére
	A kezelési költségek számításának programja
82	Programozási módszerek a feldolgozási idő rövidítésére
83	Címkezett címzés átalakítása abszolút címzésre

84	V. A készülék tulajdonságainak és feladatainak részletes elemzése
	Alapműveletek
	Alapkijelzés
	Adatbeíró billentyűk
85	Törölő műveletek
	Kettős rendeltetési billentyűk
86	Kijelzési módok
	Hatványkitevős kijelzési módok
88	Műszaki kijelzési mód
	Rögzített pontos beállítás
89	Villogó kijelzés
	Számtani műveletek
	Az alapműveletek $+$, $-$, $\times$, $\div$, $=$
90	Az algebrai műveleti rendszer
91	Zárójelek
92	„Vakműveletek” zárójelekkel
93	Algebrai függvények
	Reciprokérték $1/x$
	Logaritmusok
	A 10 és e hatványai
	Szögszámítások
	Szögszámítási üzemmódok
94	Szögfüggvények
95	Árkuszfüggvények
	Szög–radián–újfok átszámítások
96	Egészszám és abszolút érték képzése
	Négyzetre-emelés és négyzetgyökvonás
	Hatványozás és gyökvonás
97	Tárolás
	A tárolókapacitás megválasztása – a tárolómezők elosztása
98	Az adattároló törlése
	Tárolás és az adatok behívása
99	Közvetlen műveletek az adattárolókkal
	Tárolómező-kijelző csere
100	Különleges vezérlési műveletek
	Nyomatási műveletek Op 00–08
101	Egy software-program elemzése (programátvétel) Op 09
	Az előjel-függvény Op 10
	Statisztikai számítások Op 11–15
	A tárolási tartomány elosztása Op 16–17
	Vizsgálati műveletek Op 18–19
	Az adattárolók értékének növelése és csökkentése Op 20–29 (30–39)
	Átszámítások és statisztikai számítások
102	Átszámítások
103	Statisztikai számítások
	Adatbeírás
104	Középérték, szórásnégyzet és korrigált szórás

106	Lineáris regresszió
108	Trendek elemzése
109	Statisztika a számításokban
	A programozásról általában
	A számológép programozása
110	Tárolókapacitás és annak felosztása
111	Alapvető programvezérlési utasítások
112	Saját program beírása
	Saját program végrehajtása
114	Programozási műveletek
	Utasításkódok (billentyűkódok)
116	A billentyűutasítások tárolása
117	A programok módosítása
119	A programrészek megjelölése
	A programcím-billentyűk mint címkék
120	Általános címkék
	Átmeneti utasítások
121	A feltétel nélküli átmeneti utasítás és az alprogramutasítás
	Alprogramok
123	Software-programok mint alprogramok
124	Feltételes átmenetek (vizsgáló utasítások)
125	Csökkentés és átmenet 0-nál
126	Jelzők
127	Jelzők és hibafeltételek
	Közvetett címzés
129	VI. Nyomatatóvezérlés
130	Nyomtatás szükség szerint
131	Saját programok feljegyzése
132	Az adattárolók felsorolása
	A számítás jegyzőkönyvezése, párhuzamos üzem
133	Ellenőrző jelölések párhuzamos üzemmódban
134	Különleges vezérlési műveletek
135	Az adatok ábrázolása Op 07
136	A programban szereplő címkék felsorolása Op 08
137	A nyomtatófej tisztítása
	VII. Mágneskártyák
137	Rögzítés mágneskártyára
139	A programok adatvédelme
	A mágneskártyák olvasása
	Programozott kiolvasási utasítás
140	A mágneskártyák kezelése és gondozása
141	A mágnesfej tisztítókarttyájának alkalmazása
	A vizsgálókártya alkalmazása
142	A. Függelék – Karbantartás és szerviz
144	B. Függelék – Hibafeltételek
145	C. Függelék – A kijelzett eredmények és pontosságuk
146	D. Függelék – Hibakeresés a programokban

BEVEZETÉS

A mai zsebszámológépek sok területen megkönnyítik a matematika alkalmazását a feladat megoldásában; a számolás gyors, pontos, megbízható. Kapcsolatunk a számokkal és a matematikával – a rutinfeladatoktól kezdve a bonyolult problémákig – mindennaposá válik.

Kezdetben a gépi számolási lehetőség csak az alpműveletekre (összeadás, kivonás, szorzás, osztás) korlátozódott, s ez abban az időben úttörő jellegű fejlődésnek számított. A következő lépést a nagyobb teljesítményű, több függvényt tartalmazó számológépek jelentették, ezek révén lehetővé vált a négyzetreemelés, a gyökvonás, a logaritmikus és trigonometrikus műveletek gyors elvégzése. Így egyrészt sok diagram és táblázat vált fölöslegessé, másrészt lehetővé vált a műszaki gyakorlatban adódó feladatok gyors és pontos megoldása. A mai zsebszámológépek programozhatósága teljesen új területet nyit, amely eddig a számítógépek kiváltsága volt.

Ezt a kezelési útmutatót úgy szerkesztettük, hogy a felhasználó rögtön a programozással kezdhesse a tanulmányozást. Ki fog derülni, hogy milyen egyszerű valójában a programozható számológép teljesítőképességének kihasználása.

A könyv a következő módon tagolódik:

1. A bevezető rész után az alap-üzemmódokat és a programozást ismertetjük.
2. A billentyűzet és használata
3. Ezt követi a programozás részletes tárgyalása.
4. A továbbiakban ismertetünk néhány bonyolultabb programozási módszert.
5. A kézikönyv utolsó fejezete a billentyűk használatának részletes ismertetését adja, beleértve a különböző számítási lehetőségek határait. Aki ismeri a számítógépeket és a programozást, az kihagyhatja az első fejezetet és rátérhet a számológép részletes ismertetésére.

KEZELÉSI UTASÍTÁS

A készülék akkumulátorait a gyártó szállítás előtt feltöltötte. Ennek ellenére a készülék üzembehelyezése előtt, (a minden akkumulátornál fellépő önkisülés miatt) szükség lehet az újbóli feltöltésre. Ha az első bekapcsoláskor a kijelző sötét marad vagy nem megfelelően villan fel, akkor az akkumulátorokat fel kell tölteni. Ilyenkor ki kell kapcsolni a számológépet, az akkumulátortöltőhöz kell csatlakoztatni, majd néhány percig várni. Ezután már használható a gép, ugyanis a készülékkel a feltöltési idő alatt is lehet dolgozni.

Kapcsolja be a készüléket! A kijelzőn nullának kell megjelennie annak jeléül, hogy az akkumulátor fel van töltve és a számológép üzemképes. A bekapcsolással a készüléket automatikusan nullázzuk. A készülék kijelzésének vizsgálata céljából nyomjuk meg a tizedesvessző (.) és az előjelváltó (+/-) billentyűt, majd többször a 8-as számot, míg a kijelző minden helyértékén meg nem jelenik a beírt számjegy. Ügyeljünk arra, hogy a 8-as számjegy minden egyes beütésekor a tizedespont és a mínusz jel egyvel balra tolódjék! Egymás után legfeljebb 10 számjegy írható be, függetlenül attól, hogy a szám pozitív vagy negatív előjelű. A tizedik számjegy után beírt számjegyeket a készülék nem veszi tudomásul. A könnyebb olvashatóság érdekében a mínusz jel a kijelzőn közvetlenül a negatív szám előtt, baloldalt marad.

A kijelző villog, ha a számolási tartományt túlléptük, vagy ha olyan műveletet akarunk végeztetni, amire a készülék nem alkalmas. A villogás megszűnik, ha a beírást töröl CE billentyűt benyomjuk.

A készülékhez részletes utasítást adunk, de gondoljon arra, hogy az egyéni tapasztalatgyűjtést semmi sem pótolhatja! Az egyéni kezdeményezőkézség az egyik legjobb lehetőség a készülék sokoldalúságának és teljesítőképességének megismerésére. Minél több tapasztalatot gyűjt a készülék sokoldalúságáról, annál jobban tudja azt munkájában hasznosítani.

ÜZEMMÓDOK

A készülék alapvetően három üzemmódban működhet.

1. Felhasználható a félvezetővel felépített ún. software programok valamelyike. Néhány gombnyomással a készülék bonyolult feladatok megoldására készíthető elő.
2. A készülékbe saját programot is beírhatunk, melyet a gép megőriz és a kezelő kívánságára újra végrehajt.
3. A készülék mint nagyteljesítményű számológép is állandóan rendelkezésre áll bonyolult számítások elvégzésére.

A software egységek felhasználása

Az egyedi programkészítésre vonatkozó ismeretek nélkül is használhatjuk a beépített programokat. A készülék alsó részében elhelyezett úgynevezett solid-state software modul programok alapgűjteménye. Ez a cserélhető egység sok alapvető programot tartalmaz, amelynek használatát a software-egység kézikönyve részletezi. A programokat a Pgm jelű programbillentyűvel lehet a készülékbe bevinni és a leírásnak megfelelően lefuttatni.

A 21. egységben tárolt „sok-keves” játékkal mutatjuk be, milyen egyszerű ezeknek a programoknak az alkalmazása. A feladat az, hogy egy ismeretlen számot, melyet a gép választott, lehetőleg kevés számú kísérlet után kitaláljunk. A számológép által kiválasztott szám 1 és 1023 között van. A gép a kijelzőn válaszol arra a kérdésre, hogy az általunk kitalált szám azonos-e a gép által választott számmal, nagyobb vagy kisebb annál. Kezdjük tehát a játékot az alábbi programlépések alapján.

PROGRAMLÉPÉSEK

Lépés	Tevékenység	Billentyű	Kijelzés
1	A program kiválasztása	2nd Pgm 21	
2	A tizedespont, majd tetszőleges számjegyek beírása	A	a beírt szám
3	az ismeretlen szám képzése	B	0.
4	A próbálkozás beírása (1 ... 1023) a gép válaszáinak jelentése >1 = kisebb, 1 = nagyobb, villogó 0 = helyes	C	a gép válasza
5	a negyedik lépés ismétlése szükség szerinti alkalommal		
6	a lépésszám kijelzése	D	lépésszám

A legtöbb tárolt program hasonló módon, közvetlenül alkalmazható. A gyakorlatban több tucat programlépést hajtatunk végre az előre elkészített programokkal, csak azokat a számokat kell beríni, amelyekkel számolni akarunk, és el kell indítani a programot.

A solid-state software programok használatát és részleteit a programgyűjtemény kézikönyve ismerteti.

A beadott információ, a kiadott információ és a fontosabb részletek, melyekre a software-program optimális kihasználásához szükség van, mind megtalálható ebben a kézikönyvben. Ügyeljünk tehát az utasításokra, ha tárolt programokkal dolgozunk! Ajánlatos az alprogram-gűjteményt azonnal átnézni, így megismerkedhetünk azokkal a programokkal, amelyek a rendelkezésünkre állnak, és a feladat megoldására rögtön felhasználhatók.

Számolás a billentyűzet segítségével

A FTK 1096 korszerű számológép, az információ beírásakor az ún. AOS algebrai logikát használja. A gép a számokat jobbról balra, a műveleteket pedig az algebrai írásmódnak megfelelően balról jobbra haladva végzi. Ha pl. a 100°C és a 36°C és a -4°C hőmérsékleti értéket Fahrenheit-fokra akarjuk átszámítani, akkor a Celsius-értékeket $\frac{9}{5}$ -del meg kell szorozni és ehhez 32 -t kell hozzáadni. $F^{\circ} = C^{\circ} \cdot \frac{9}{5} + 32$.

Billentyű	Kijelzés
100 X	100.
9 ÷	900.
5 +	180.
32 =	212.

Ugyanígy módon kapjuk a $36^{\circ}\text{C} = 96,8\text{F}^{\circ}$ és a $-4^{\circ} = 24,8\text{F}^{\circ}$ azonosságot. Az AOS logikát és az ebből adódó számolási lehetőségeket a könyv későbbi fejezete tárgyalja.

Példa saját program készítésére

Ha kialakult egy számolási módszer, amely segítségével többféle kiinduló adattal akarjuk az eredményt kiszámítani, nyomjuk meg az LRN jelű — learn — (tanulás) billentyűt, és tápláljuk be a szóbanforgó számítási módszert, azaz programot a számológépbe! Az előbbi példát véve alapul, először az LRN billentyűt nyomjuk meg, majd utána a következőket: $X \ 9 \div 5 + 3 \ 2 = R/S$ (programleállítás és eredménykijelzés). Ezután ismét nyomjuk meg az LRN billentyűt, miáltal a számológép befejezi a betápláltak „megtanulását” és készen áll arra, hogy a műveletsort bármely beadott számjeggyel megismételje.

A teendők a következők

1. Először beírjuk a Celsius-értéket
2. Ezután a gépet az RST (reset) billentyűzéssel utasítjuk, hogy a program kezdetére álljon.
3. Végül megnyomjuk az R/S (run/stop) billentyűt és ezzel elindítjuk a programot.

Billentyű	Kijelzés
100 RST R/S	212.
36 RST R/S	96.8
4 +/- RST R/S	24.8

Ez a szükséges összes tevékenység a $C^{\circ} - F^{\circ}$ átszámításához. Ilyen egyszerűen állíthatunk elő saját programot; ez a számológép alapvető tulajdonsága. Amennyiben egy programot rögzítettünk, helyességét és pontosságát felülvizsgáltuk, akkor azt egyetlen gombnyomással bármikor felhasználhatjuk.

NYOMTATÁSI LEHETŐSÉGEK

A számológéphez KA-100 típusú nyomtató is csatlakoztatható. A gép utasítást ad a nyomtatónak a kijelzett értékek papírra nyomtatására. Ha a feladatot kézi billentyűzéssel oldjuk meg, akkor tetzés szerint akár egy-egy, akár valamennyi részeredmény kinyomtatható, vagy egy tárolt program teljes listája kiírható. Ahogy a program fut, a nyomtatási parancsok alapján automatikusan kiírhatók a kijelzőben levő értékek. Ez lehetővé teszi, hogy a program lefuttatásával egyidejűleg egy-egy sor részeredményt is kinyomtassunk. A „trace”-üzemmódban (a párhuzamos üzemmód) a nyomtató valamennyi lépést a hozzá tartozó számadattal kinyomtatta.

Különleges vezérlő műveletek alkalmazásával olyan információkat lehet összeállítani és kinyomtatni, melyek alkalmasak arra, hogy egyes programszakaszokat vagy programcímeket megjelöljünk. Soronként 20 darab írásjel nyomtatható ki; amelyek 64-féle írásjel közül választhatók ki.

SZÁMOLÓGÉPES RENDSZER FELÉPÍTÉSE

A számológép sok nagyteljesítményű egységből áll, melyek összességükben egy feladatmegoldó rendszert képeznek. Ebben a fejezetben rövid összefoglalást adunk a rendszer néhány tulajdonságáról. A következő fejezetben áttekintést adunk a különböző rendeltetésű egységekről.

II. BILLENTYŰZET

Mielőtt a készülék tulajdonságaival részletesebben foglalkoznánk, célszerű röviden áttekinteni a billentyűzet jellegzetességeit és szerepét. Ez mindennek előtt akkor indokolt, ha a felhasználó először ismerkedik korszerű számológéppel.

A BILLENTYŰZET FONTOSABB TULAJDONSÁGAI

A kijelzés törlése CE, CLR

Ez a két billentyű törli a kijelzőregisztert, a feladatmegoldással kapcsolatos követelményekről függően.

A beírt információ törlése: — clear entry (CE)

A műveleti tárolót törölő billentyű törli a kijelzőn látható, utolsónak beírt számot, feltéve, hogy közben semmilyen műveleti vagy függvény billentyűt nem működtettünk. A CE billentyűnek nincs befolyása az éppen folyó számításra. (Ha pl. számjegybeírás során 5 helyett 6-ot nyomtuk meg, akkor először a CE jelű billentyűvel töröljük, majd beírjuk a helyes számot.) A kijelzőnek valamilyen hiba esetén fellépő villogását is a CE-billentyűvel szüntethetjük meg.

Törlés — clear — (CLR). Ez a törölőbillentyű a kijelzőregiszter tartalmával együtt a folyamatban levő számítás is törli. Ha előzőleg hibajelzés is volt, ezt is törli.

Adatbeíró billentyűk 0 ... 9, +, -, ×, ÷, π

Számjegyeket a 0 ... 9, a +, - és a π billentyűk segítségével írunk be. Ha egy számjegyet beírunk, a tizedespontra a beírt számtól jobbra helyezkedik el, mindaddig, amíg a tizedespontra egyet balra lép a törtrészrel együtt. Ha a kijelzőn egy szám előjelét meg akarjuk változtatni, nyomjuk meg egyszer a +/- jelű billentyűt! Második megnyomásra az előjel természetesen ismét az eredetire változik. A 2nd és a π jelű billentyű ilyen sorrendben való megnyomásával a kijelzőn megjelenik a π első tíz számjegye (3.141592654), a belső kijelzőregiszterben pedig a szám 13 számjegyét tárolja a gép (3.141592653590). Ezt a beírást nem törölhetjük a CE billentyűvel.

Az alapműveletek billentyűi +, -, ×, ÷, =.

Az aritmetikai műveleteket a fenti öt (+, -, ×, ÷, =) billentyűvel lehet elvégezni. A számológép a nagyteljesítményű AOS típusú beírási módszerrel dolgozik, így a megoldási mód nagyon egyszerű. A feladatot úgy írjuk be a gépbe, mintha papírra írnánk, majd az = billentyűzésénél megkapjuk a végeredményt.

Az AOS rendszerű beírás előnye, hogy az összetett műveleteknél a gép automatikusan biztosítja a helyes műveleti sorrendet. Így az eredmény ennek megfelelően alakul. Az AOS-módszert a következő fejezet részletesen tárgyalja. Az = jelű billentyűvel a függőben levő műveleteket — azokat is, amelyeket a gép maga végez — zárjuk le. A végeredmény birtokában a beírást töröljük és hozzáfoghatunk az újabb feladathoz.

Pt. $15 + 7 \times 31 - 4 = ?$

Billentyűzés: $15 + 7 \times 31 - 4 =$ kijelzés: 228.

Megjegyzés: tartsuk szem előtt, hogy az AOS módszer alapján a gép a fenti kifejezést, mint $15 + (7 \times 31) - 4$ műveletsort értelmezi. Vagyis, először 7×31 értéket számolja, utána 15-öt hozzáad, majd ebből 4-et levon.

Az AOS beírási módszer

A matematika egyértelműen meghatározott törvényekre épülő tudomány. E törvények egyike kimondja, hogy egy és ugyanazon műveletsornak ugyanazokkal az adatokkal sohasem lehet különböző eredménye. Általános szabályok szolgálnak a vegyes műveleteket tartalmazó feladatok egyértelmű megoldására.

Pl. a következő feladatnak csak egy helyes megoldása van:

$$3 + 10 - 2 \times 14 \div 7 = ? \quad \text{Az eredmény 9.}$$

A feladatot közvetlenül balról jobbra billentyűzzük, így kapjuk meg a helyes választ. A számológép rangsorolja a beírt műveleteket, mégpedig a következő sorrendben, ill. elvek szerint:

1. Egyváltozós függvények billentyűi. A gép billentyű-benyomása után azonnal végrehajtja ezeket a kijelzett számra vonatkozóan. Ide tartoznak a szögfüggvények és a logaritmusfüggvények, ezek inverzei, valamint a négyzetreemelés, négyzetgyökvonás, a reciprokszámítás és az átszámítási műveletek. Ezekről a billentyűkről részletesebb tájékoztatást adunk a jelen fejezet további részében.
2. Sorrendben ezután következnek a hatvány- és gyökműveletek (y^x és $\sqrt[y]{y}$). (Ezeket a fentiekhez hasonlóan még ebben a fejezetben részletezzük.)
3. Ezt követi a szorzás és az osztás.
4. Végül az összeadás és kivonás.

A fenti algebrai rangsor minden zárójelben levő kifejezésre érvényes. Végezetül az = jelű billentyű zárja le a műveletsort.

Természetesen előfordul az is, hogy a kezelő maga akarja meghatározni a műveleti sorrendet. Ilyen esetekben a sorrend a () zárójel billentyűvel befolyásolható (l. a következő részt). A zárójelek különös jelentőséggel bírnak a matematikában, és erre tekintettel voltak a számológép tervezésénél is.

A zárójel-billentyűk (,)

Sok esetben magunknak kell egy-egy kifejezés számításának műveleti sorrendjét meghatározni, ilyenkor meg kell adni a számok csoportosítását, mely a megoldáshoz szükséges. A zárójelek adják meg azt a lehetőséget, hogy a számokat és műveleteket csoportosítsuk. Ha egy sor számot és műveletet zárójelbe teszünk, utasítjuk a gépet arra, hogy először ezt a részfeladatot oldja meg, egyetlen számértékre egyszerűsítve és ezt a részeredményt illeszse be a hátralevő számítások sorrendjébe. A zárójelen belül a gép a megadott rangsorolás szerint dolgozik. Célzerű a zárójelek használata akkor is, ha kétségeink vannak afelől, hogy milyen sorrendben fogja a gép a betáplált műveleteket végrehajtani. A gép 9 nyitó zárójel fogadására alkalmas; vagyis egyidejűleg maximum nyolc befejezetlen művelet lehet a számológépben.

$$(((2 \times (2 \times (2 \times (2 \times (2 \times (2 + 2y \times (2 + 2)) - (2 + 2)))))) \div 2) \div 2)$$

Ennek a műveletsornak a beírásakor ügyelni kell arra, hogy a gép addig nem végez műveletet, míg az első zárójelet be nem írjuk. A gép tárolja az összes utasítást, és azokat a megfelelő időpontban hajtja végre.

Még egy fontos dologra kell ügyelni a zárójelek használatakor. Gyakran láthatunk olyan egyenleteket vagy kifejezéseket, amelyben a zárójelek szorzásra utalnak, anélkül, hogy ez külön jelölve lenne. Pl. $(2+1)(3+2) = 15$. Ebben a kifejezésben a számológép nem végzi el a szorzást. A műveleti jelet minden esetben ki kell írni a zárójelek közé:

$$(2 + 1) \times (3 + 2) = 15.$$

Egy példa a zárójelek alkalmazására: számítsuk ki a következő kifejezést!

$$\frac{8 \times (4 + 9) + 1}{(3 + 6 \div 2) \times 7}$$

Ilyen feladatoknál a gépnek először ki kell számítania a számlálót, majd osztani kell a kiszámított nevezővel. Hogy biztosak legyünk a helyes számítási sorrendben, a feladat beírásakor a számlálót is és a nevezőt is külön tegyük zárójelbe. (Írásban a törtvonal helyettesíti ezt a két zárójel-párt.)

Billentyű	Kijelzés	Megjegyzés:
CLR	0	a korábbi számítások törlése
($8 \times (4 + 9)$	13.	a (4+9) kiszámítása
+	104.	a $8 \cdot (4+9)$ kiszámítása
)	105.	a számláló értéke
$\div (3+6 \div 2)$	6.	a $(3+6 \div 2)$ értéke
$\times 7$	42.	a nevező értéke
=	2.5	végeredmény

Kettős rendeltetésű billentyűk 2nd, INV

A számológépnek számos olyan előnye van, amellyel időt takaríthatunk meg és a számítások pontosságát növelhetjük. Hogy ne legyen sok billentyűre szükség, a legtöbb billentyűvel egynél több művelet végezhető. A billentyű elsődleges rendeltetése közvetlenül a billentyűre van írva. Ezt úgy használjuk, hogy egyszerűen megnyomjuk a szóbanforgó billentyűt. A másodlagos rendeltetést — melynek jele közvetlenül a billentyű fölé van nyomtatva — akkor végzi a gép, ha előzőleg működtetjük a 2nd jelű billentyűt.

Pl. az $\ln x$ billentyű megnyomásával a természetes logaritmust, a 2nd és az $\ln x$ billentyű (a továbbiakban 2nd log) lenyomásával a tízes alapú logaritmust számítjuk ki.

Az INV jelű inverzió-billentyű révén is növekszik a gép funkcióinak száma, a billentyűk számának növelése nélkül. Ha az INV billentyűt valamely másik előtt működtetjük, akkor az utóbbinak az inverz függvényt számítja ki a gép. Az INV és a 2nd billentyű segítségével a 108 különböző művelet mindössze 45 billentyűvel hajtható végre. A különleges billentyűk alkalmazásáról további ismertetést az V. fejezet Kettős műveleti billentyűk c. alfejezete tartalmazza.

Tárolóbillentyűk CMs, STO, RCL, Exc.

A készülék bekapcsolásakor 60 adattároló áll rendelkezésre. Az adattárolók száma azonban a meglevő programtároló befogadóképességének terhére változtatható. Az ezzel kapcsolatos részleteket a Tárolási lehetőségek c. fejezetben a Tárolási kapacitás megválasztása c. alfejezet tartalmazza. Az adattárolók a gép azon különleges részei, amelyek későbbi felhasználás céljára adatokat tárolnak. Minthogy általában egynél több adattároló áll rendelkezésre, ezért a felhasznált tárolót kétjegyű címmel kell megjelölni. Pl. STO 08.

A CE és a CLR jelű billentyű nincs hatással az adattároló tartalmára. A 2nd CMs írással az összes adattárolót egyidejűleg törölhetjük, azaz a regiszterek tartalma 0 lesz.

STO XX-tárolás (store). Ezzel az utasítással a kijelző tárolójában levő számot az XX = 00...99 adattárolóban tárolhatjuk anélkül, hogy a kijelzőregiszter tartalmát befolyásolnánk. A kérdéses regiszterben előzetesen tárolt számjegyet a gép egyidejűleg törli.

RCL XX-visszahívás (recall). Ez az utasítás az XX adattároló tartalmát ismét a kijelző tárolójába viszi, az ott levő szám helyébe. Az XX adattároló tartalma megmarad.

2nd Exc XX-tárolócsere (memory exchange). Ezzel a művelettel az XX adattároló és a kijelző-tároló tartalmát cseréljük fel. A kijelző tárolójában levő számot az XX tárolóba visszük, miközben a kijelző tárolójába az XX adattároló tartalma kerül. Ez a billentyű sok esetben igen hasznos; ha az adattároló tartalmát kívánjuk felhasználni anélkül, hogy a kijelző tárolójában levő értéket elvesztenénk.

Példa: a 3,21 szám tárolása és visszahívása.

Billentyű	Kijelzés	Megjegyzés
3.21 STO 08	3.21	3.21 értéket a 08 adattárolóban tároljuk
CLR	0	a kijelzés törlése
RCL 08	3.21	a 08 adattároló tartalmának visszahívása.

Példa: számítsuk ki a következő kifejezést:

(A+2) + A (A+2), ha A = 9.3069128

Billentyű	Kijelzés	Megjegyzés
CLR	0	törlés
9.3069128 STO 12	9.3069128	az A értékét a 12 adattárolóba visszük
+ 2 +	11.3069128	az A+2 számítás
2nd Exc 12	9.3069128	az (A+2)-t bevisszük a 12 adattárolóba és az A értékét visszahívjuk a kijelzőbe
X RCL 12	11.3069128	az A+2 visszahívása a kijelző tárolójába. Ügyeljünk arra, hogy az A és az A+2 között az X billentyűt is működtessek!
=	116.5393643	Az összes függőben levő műveletet befejezzük, az eredmény a kijelzőn látható.

A sokjegyű számot csak egyszer kell beírni, így módon időt takarítunk meg, és a billentyűzési hibákat elkerüljük. A tároló-cserélő billentyű biztosítja a tárolás és a visszahívás elvégzését, ami egyszerűsíti a számítást.

Műveletek az adattárolókban SUM, Prd

Számos művelet elvégezhetünk magában az adattárolóban, a folyamatban levő számítások zavarása nélkül.

SUM XX-összeadás a tárolóban (memory sum). Ezzel a művelettel a kijelző tárolójának tartalmát közvetlenül hozzáadjuk az XX adattároló tárolt értékhez. Az összeadás eredményét a megnevezett adattároló tárolja anélkül, hogy a kijelző tartalma megváltoznék. Hasonlóan működik az INV SUM XX billentyűcsoport, amikor is a kijelző tárolójának értékét az XX adattároló tartalmából kivonjuk.

2nd Prd XX szorzás a tárolóban (memory product). Ezzel a művelettel az XX adattároló tartalmát a kijelző tárolójának értékével szorozzuk meg. Az INV 2nd Prd XX billentyűcsoporttal az XX adattároló értékét a kijelző tárolójának értékével osztjuk el. Az eredményt ismét tároljuk, míg a kijelző tárolója változatlan marad. Az adattárolóban végzett számítás hasonlít a billentyűzettel végzett számításához, azzal a különbséggel, hogy az eredmény nem a kijelző tárolójában, hanem az adattárolóban jelentkezik.

Példaként számítsuk ki a 28 Ft-os és a 6,60 Ft-os tétel összköltségét 5% adó esetén!

Billentyű	Kijelzés	Megjegyzés
28 STO 01	28.	28 a 01 adattárolóba
6.6 SUM 01	6.6	6,6-ot hozzáadjuk a 01 adattároló tartalmához
1.05 2nd Prd 01	1.05	a 01 adattároló tartalmát 1,05-tel megszorozzuk
RCL 01	36.33	összköltség

A KIJELZÉS VEZÉRLÉSE

Alap-kijelzési mód

A kijelzés számjegyes információt szolgáltat, beleértve az előjelet, a tizedesponstot; továbbá villogással jelzi a hibát és a túlsordulást. (A B függelék tartalmazza azokat a feltételeket, amelyek mellett a kijelző hibát jelez.) A beírt szám legfeljebb 10 jegyű lehet. A tizediket követő számjegyeket a gép nem veszi figyelembe.

lebegőpont

negatív előjel - 9076.321445

egész számok tizedes értékek

A kijelző és a kijelzőtároló fogalma nem azonos. A kijelző csak az optikai kijelző berendezésre vonatkozik. A kijelzőtároló egy belső tárolót jelent, amely az eredményeket 13 helyen képes fogadni. Ha $x < 10^{-10}$ vagy $x \geq 10^{10}$, akkor a készülék automatikusan áttér az exponenciális kijelzési formára. Ha pl. 400 000 és 2 000 000 szorzatát képezzük, akkor az eredményül kapott 800 000 000 000 túlszordulást okozna a 10 jegyű kijelzőn. Ehelyett a kijelzés 8.11 lesz (8.11 megfelel $8 \cdot 10^{11}$ -nek).

8. 11

Exponenciális kijelzési mód EE

Sok területen — különösen a tudományokban és a technikában — gyakran kell igen kis vagy igen nagy számokkal számolni. Ezek a számok exponenciális alakban minden nehézség nélkül feldolgozhatók. Bármely szám exponenciális alakban egy alapszám (mantissa) és 10 valamely hatványának szorzata (normálalak).

(Megjegyezzük, hogy a matematika a mantissa kifejezést eltérő értelemben használja: a mantissa az alapszám logaritmus!)

Szám = mantissa $\cdot 10^{\text{karakterisztika}}$

Beírás exponenciális alakban:

1. Beírjuk a mantisszát, legfeljebb 8 jegyű számmal (nyomjuk meg a +/- billentyűt, ha a mantissa negatív),
2. az EE billentyűzésekor jobboldalt a kijelzőn 00 jelenik meg,
3. beírjuk a karakterisztikát és a +/- billentyűt működtetjük negatív kitevő beírása esetén.

Egy szám, pl. $-3.8901448 \cdot 10^{-32}$ így jelenik meg a kijelzőn.

-3.8901448 - 32

Exponenciális alakban a kitevő mutatja, hogy hová kerülne a tizedespon, ha a számot kiírnánk. Pozitív kitevő azt mutatja, hogy hány hellyel kellene a tizedesponot balra, negatív kitevő azt, hány hellyel kellene jobbra eltolni.

Példa: $2.9979 \cdot 10^{11} = 299\,790\,000\,000$. (A tizedespon 11 hellyel jobbra tolódott, és ennek megfelelő számú 0-val kellett a számot kiegészíteni).

Más példa: $1.6021 \cdot 10^{-9} = 0.0000000016021$. (A tizedespon 9 jeggel balra tolódott és a számot ennek megfelelően kellett 0-kal kiegészíteni.) Ha egyszer az exponenciális alakot választottuk, az mindaddig megmarad, míg az INV EE billentyűk megnyomásával vissza nem térünk az alakjelzési módra, a CLR a kijelző törlésével egyidejűleg ezt a kijelzési módot is megszünteti.

Műszaki kijelzés Eng

A műszaki kijelzés az exponenciális kijelzési mód egy változata. A kitevőben csak a három többszöröseként előállítható szám szerepelhet (10^{12} , 10^6 stb.) A mantissa egy, két vagy három jegyű lehet. Így lehetőség van arra, hogy olyan nagyságrendekben számoljunk, ami a tudósok, mérnökök és technikusok által használt mértékegységeknek leginkább megfelel (10^{-12} a pikofarádnak, 10^{-3} a milliméternek, 10^3 a kg-nak vagy 10^6 a mikroszekundumnak az együtthatója). A kijelzőt a 2nd Eng billentyűzéssel lehet műszaki kijelzésre állítani. Ezt a kijelzési forma az INV 2nd Eng billentyűcsoporttal törölhető. Példaként számítsuk ki a $8 \cdot 98 \cdot 30$ szorzatot műszaki kijelzéssel!

Billentyű:	Kijelzés:
CLR 2nd Eng	0. 00
8 X 98 X	748. 00
30 =	23.52 00
INV 2nd Eng	23520.

Meghatározott számú tizedesjegy beállítása Fix

Sok esetben előnyös lehet, ha előre megválaszthatjuk, hogy hány számjegyből álljon a tizedestört. A 2nd Fix és egy 0-tól 8-ig terjedő szám beírásával dönthetjük ezt el. A számológép a továbbiakban ennyi tizedesre kerekíti a kijelzett számot, de csak a kijelzőben. Beírni viszont továbbra is annyi helyiértékkel lehet, amennyi szükséges, mert a gép saját belső, 13 helyiértékű pontossággal tárol. A rögzített helyű tizedespont az INV 2nd Fix billentyűcsoporttal oldható fel.

Pl. $2:3 = 0.666666667$

Billentyű:	Kijelzés:
CLR	0.
$2 \div 3 =$	.666666667
2nd Fix 6	0.666667
2nd Fix 2	0.67
2nd Fix 0	1.
INV 2nd Fix	.666666667

Bekapcsoláskor a gép alap (lebegőpontos) kijelzésű.

MATEMATIKAI FÜGGVÉNYEK

Négyzet, négyzetgyök és reciprok érték x^2 , $\sqrt{x}$, $1/x$

Ez a három billentyű igen fontos az egyenletek gyors megoldásánál. Mindhárom billentyű közvetlenül a kijelzőtárolóban levő számjegyre hat anélkül, hogy a folyó számításokat zavarná.

x^2 a kijelzőben levő szám négyzetreemelése

$\sqrt{x}$ a kijelzőregiszterben levő szám négyzetgyökének számítása

$1/x$ a kijelzőregiszterben levő számmal osztjuk az 1-et

Példa a három művelet használatára:

Billentyű:	Kijelzés:	Megjegyzés:
CLR	0	törölés
$4 \sqrt{x}$	2	$\sqrt{4}$
$\div 5 \ 1/x$	0.2	$1/5$
x^2	0.04	$(1/5)^2$
=	50.	végeredmény

Hatványok és gyökök y^x

Ezzel a billentyűvel bármely pozitív számot tetszés szerinti hatványra emelhetünk, és bármely pozitív számból tetszés szerinti gyököt vonhatunk.

A hatványozás lépései y^x

1. a hatványozandó szám (y) beírása
2. az y^x billentyűzése
3. a hatványkitevő (x) beírása
4. az = vagy bármely másik műveleti billentyű megnyomása

Példa: 2^6 számítása.

Billentyű:	Kijelzés:
CLR	0
$2 \ y^x \ 6 =$	64.

A gyökvonás lépései INV, y^x

1. az y szám beadása, melyből gyököt akarunk vonni.
2. az $\text{INV } y^x$ billentyűk megnyomása
3. a gyökkitevő (x) értékének beírása
4. az = vagy akármilyen másik műveleti billentyű megnyomása.

Példa: $\sqrt[6]{64}$ számítása.

Billentyű:	Kijelzés:
CLR	0
$64 \ \text{INV } y^x \ 6 =$	2.

Megjegyzés: az y értéke csak pozitív lehet, negatív értéknél villog a kijelző.

Logaritmusok $\ln x, \log$

A logaritmusok olyan matematikai függvények, amelyeket sok műszaki és elméleti számításnál alkalmaznak. Általánosságban igaz, hogy ha $x=y^2$, akkor $\log_y x=2$. A következőkben leírt billentyűkkel logaritmust lehet számítani, logaritmustáblázatok lapozgatása és egyéb számítások nélkül.

$\ln x$ – természetes logaritmus. Ezzel a művelettel a kijelzőregiszterben levő szám természetes logaritmusát (alapja az $e=2.718281828459$) számíthatjuk ki. Ha a szám negatív vagy nulla, a kijelző villog. A logaritmus-értékből a szám visszakeresése – a numerus logaritmhi – természetes alap esetén e^x , az $\text{INV } \ln x$ billentyű-zéssel lehetséges.

$2\text{nd } \log$ – tízes alapú logaritmus. A kijelzőregiszterben levő szám tízes alapú logaritmusát számíthatjuk ki. A számnak itt is pozitívna kell lennie. A logaritmusértékből a szám (10^x) visszakeresése az $\text{INV } 2\text{nd } \log$ billentyűzéssel lehetséges.

Példaként számítsuk ki az $(e^{2.7} + 10^{1.2})$ szám természetes logaritmusát!

Billentyű	Kijelzés:	Megjegyzés:
CLR	0	törlés
($2.7 \ \text{INV } \ln x$	14.87973172	az $e^{2.7}$ számítása
+ $1.2 \ \text{INV } 2\text{nd } \log$	15.84893192	a $10^{1.2}$ számítása
)	30.72966365	a befejezetlen összeadás elvégzése
$\ln x$	3.425195888	végeredmény

Szögérték-billentyűk Deg, Rad, Grad

A számológéppel sokféle szögszámítás végezhető. Ilyenek elsősorban a trigonometrikus függvények és a poláris-derékszögű koordináta-át számítások. A szögeket (rég) fokban, újokban (grádban) vagy ívmértékben adhatjuk meg. Bekapcsoláskor a gép fokban számol. A fenti billentyűk kiválasztásával azonban áttérhetünk más szögegységre.

Felhívjuk a figyelmet arra, hogy a szögérték-billentyű lenyomásakor a készülék a szögegységet változtatja meg; a szögértéket nem számítja át!

2nd Deg számolás (régí) fokban. Ebben az üzemmódban az összes szögértékeket fokban írjuk be és számíjuk ki. 1 (régí) fok a körnek 1/360 része, a derékszög 90 foknak felel meg.

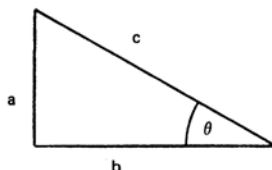
2nd Rad számolás ívmértékben (radián). Ebben az üzemmódban az összes szöget radiánban számíthatjuk.

(1 radián = $\frac{360^\circ}{2\pi}$, a derékszög $\frac{\pi}{2}$ rad.)

2nd Grad számítás újfokban. Ebben az üzemmódban az összes szöget újfokban kell megadni. 1 újfok a körnek 1/400-része, a derékszögnek 100 újfok felel meg).

A trigonometrikus függvények billentyűi sin, cos, tan

Ezekkel a billentyűkkel a kijelző tárolójában levő szögérték szinusz, koszinusz, ill. tangense közvetlenül számítható. A szög értékét az előzőekben leírt háromféle módon lehet megadni.



$$\sin \theta = \frac{a}{c} \quad \cos \theta = \frac{b}{c} \quad \tan \theta = \frac{a}{b}$$

Az árkuszfüggvények az INV sin, az INV cos és az INV tan billentyűcsoportokkal számíthatók. A gép a szögértékeket a megválasztott egységben jelzi ki. Ha szögegységnek a régi fokot választjuk, akkor a szöveget decimális formában kell megadni. (Lásd még az V. fejezetet!)

ÁTSZÁMÍTÁSOK

Régi fok átszámítása D. MS

A szögnek régi fokokban való megadására két mód kínálkozik. Az egyik módszer szerint a szöveget decimálisan DDD.dd alakban adjuk meg. Itt DDD az egész fokot, a .dd a tört értékű fok tizedes törtben kifejezett értékeit mutatja. Az utóbbi helyiértéke legfeljebb 10-ig terjedhet. A másik módszer szerint a szög értékét fokban, percben és másodpercben, DDD.MMSSss alakban adjuk meg. Itt DDD a fokot, MM a percek, SS a másodpercet jelenti. Nagyobb pontossági kívánalmak esetén még a másodpercek törtrészei is megadhatók. Ügyeljünk arra, hogy a tizedesvessző a szögértéket és a perceket választja el! A fok-perc-másodperc alakról a decimális alakra való átszámításhoz írjuk be a szöveget, (DDD.MMSSss) és nyomjuk meg a 2nd D. MS billentyűket!

Az INV 2nd D. MS billentyűkkel a decimálisan megadott szögértéket átszámíthatjuk fok-perc-másodperc-re. A percek és másodpercek részére mindenkor két helyértéket kell igénybe venni, mert a gép a tizedestört beadása után mindig egyidejűleg két-két helyet vesz figyelembe; éppen ezért az utolsó nullákat nem kell beírni. Példaként először alakítsuk át az $54^\circ 02' 09.6''$ -et decimális alakra, majd vissza!

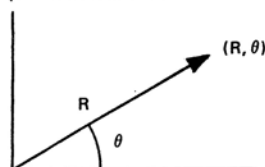
Billentyű	Kijelzés	Megjegyzés
54.02096 2nd D. MS	54.036	DD.ddd
INV 2nd D. MS	54.02096	DD.MMSSss

Ugyanez az eljárás alkalmazható az óra-perc-másodperc értékek decimális alakra való átszámítására és viszont.

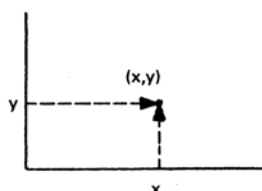
Poláris és derékszögű koordináták átszámítása P → R

A számológépnek ez a tulajdonsága különösen előnyös tudományos és műszaki területen. Az $x \approx t$ jelű billentyű, éven különösebb fáradtság nélkül átszámíthatók a poláris-koordináták derékszögűre és viszont.

poláris koordinátákról



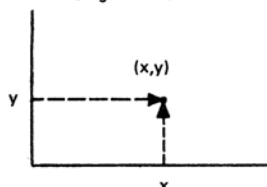
derékszögű koordinátákra



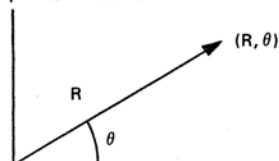
1. az R érték beírása
2. az $x \leftrightarrow t$ billentyű megnyomása
3. a θ értékének beírása (ügyeljünk a szövegységre!)
4. a 2nd $P \rightarrow R$ billentyűk megnyomásával kapjuk meg az y értékét.
5. az $x \leftrightarrow t$ billentyűzésével megkapjuk az x értékét.

A derékszögű koordináták átszámítása poláris koordinátákra a következők szerint történik:

derékszögű koordinátákról

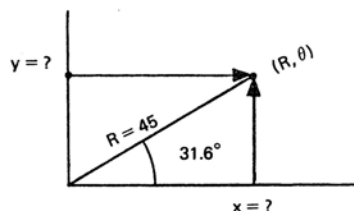


poláris koordinátákra



1. az x értékének beadása,
2. az $x \leftrightarrow t$ billentyűzése,
3. az y beírása,
4. az INV 2nd $P \rightarrow R$ billentyűzés a választott szövegységekben való kijelzésre,
5. az $x \leftrightarrow t$ billentyűvel az R értékének kijelzése.

Számítsuk át az ábra szerint az $R = 45$ m, $\theta = 31,6^\circ$ polárkoordinátákat derékszögű rendszerre!



Billentyű
CLR 2nd Deg

Kijelzés
0

Megjegyzés
törlés, szövegység-választás
régi fokban

45 $x \leftrightarrow t$

0

az R értékét a t-tárolóban tároljuk
a szögérték beírása

31.6

31.6

átszámítás derékszögű koordinátákra és
az y értékének kijelzése

2nd $P \rightarrow R$

23.57936577

az x értékének kijelzése (y most a t-tároló-
ban van.).

$x \leftrightarrow t$

38.32771204

Megjegyzés: ennél az átszámításnál egy különálló tárolót, az ún. t-tárolót használjuk az $x \leftrightarrow t$ jelű (x -t cserét végző) billentyűvel. Ezen tároló alkalmazására vonatkozó lehetőségeket a különböző programozási fejezetekben tárgyaljuk.

Középérték, szórásnégyzet, szórás $\bar{x}$, $0p\ 11$, $0p\ 11\ \sqrt{x}$

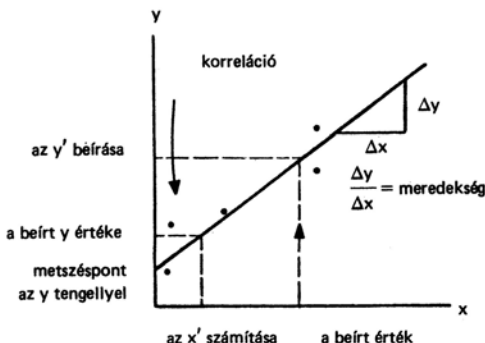
Akkor van szerepe, ha olyan adatszoportokat kell feldolgozni, amelyekben az adatok egy bizonyos változó vagy tényező körül ingadoznak. Ezek az adatok lehetnek kereskedelmi számadatok vagy vizsgálati eredmények. Statisztikai számítások segítségével ezek az adatok néhány jellemző számértékre egyszerűsíthetők, mint középérték, szórásnégyzet és szórás. A középérték az adatok átlagértéke. A szórásnégyzet és a szórás arról ad felvilágosítást, mennyire térnek el az adatok ettől az értéktől.

Részletes felvilágosítást ezen függvények alkalmazásáról az V. fejezet Statisztika című alfejezetében találhatunk.

Lineáris regresszió 0p 12...15

A lineáris regresszió fogalma nagyon tudományosan hangzik, de a számítási eljárás ezzel a géppel rendkívül egyszerű. Célja a jövőbeni események előrejelzése.

Lineáris regresszió számításánál az adatok általában összetartozó pontpárokkal jellemezhetők, melyek diagramban is ábrázolhatók. Egy pontot általában az x , y értékpárral adunk meg (lehet pl. x a hirdetési költség, y a forgalmat jellemző szám, de lehet x egy vizsgálati eredmény, y pedig egy adott x -hez tartozó teljesítménycsúc egy bizonyos területen stb.) Szeretnénk megtudni, hogy egy választott x értékhez milyen y érték (vagy fordítva) várható az előzmények alapján. A számológép úgy oldja meg a feladatot, hogy megkeresi az adatpontok közé legjobban illeszkedő egyenest. Ez az egyenes használható előrejelzés céljára. Mielőtt beírtuk az adatokat, a gép a következő információt adja ki.



A géppel hasonló módon lehet statisztikai számításokat végezni akkor is, ha a legjobban illeszkedő görbe nem lineáris, hanem pl. exponenciális.

A gép ilyen tulajdonságait az V. fejezet statisztikára vonatkozó része ismerteti részletesen.

III. A „BEÉPÍTETT” PROGRAMOK ALKALMAZÁSA

Bevezetés a szilárdtest-áramkörű software-programokba

A software fogalomnak igen sok meghatározása ismeretes. Lényegében utasításokat, programokat értünk alatta, amelyek valamilyen képlettel írhatók be. Ezek utasítják a számológépet különféle, a feladattól függő művelet sor végrehajtására. A számológépnek célszerű és egyszerűen alkalmazható programválasztéka van, amely a gépbe betehető és egyetlen gombnyomással felhasználható. Ezek a programok, amelyek tárolása a gép alsó rekeszében elhelyezett program modulban történik, a célszerű alkalmazás szempontjait figyelembe véve, számos szakterületre készültek. Az egységek egyszerűen cserélhetők. A programinformációt egy igen ismertető félvezető szilícium-morzsa (ún. chip) tartalmazza, melynek felépítése hasonló a számológép központi vezérlőegységében levő, szilíciumalapú integrált áramkörökhöz. Innen származik a félvezető szilárdtest (solid-state) software program elnevezés. A solid-state software modul sok software-programot tartalmaz. Előnyei:

1. kis helyen nagy programkapacitás, ami könnyen szállítható és alkalmazható
2. a software-programok a billentyűzet segítségével könnyen hozzáférhetők
3. a software-programok úgy készültek, hogy azokat a kezdők is nehézség nélkül használhassák.

A software-egység

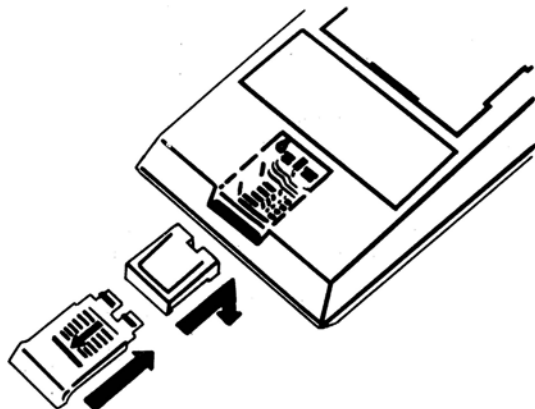
A készülék aljában helyezkedik el a „master 1” megjelölésű software-egység, amely 25 programot tartalmaz. Az egység hosszú élettartamú, de a tárolt programok épsége érdekében gondosan kell kezelni. A gyártó cég az egységek cseréje során bekövetkezett meghibásodásért nem vállal felelősséget.

Vigyázat! Az egységek cseréjekor testünk ne legyen statikusan feltöltve!

Ez különösen akkor fontos, ha a számológéphez töltőkészüléket csatlakoztattunk, mert ez a gépet leföldeli. Egyszerűen érintsünk meg egy fémtárgyat, hogy a statikus töltést testünkről levezessük. Az egységben levő információk a statikus kisülés révén súlyosan megsérülhetnek. A modulok karbantartásáról részleteket az A. függelék tartalmaz. Az alapprogramot tartalmazó egységet a gyártó a készülékbe beépíti, de az könnyen kicserélhető. Az egységet ajánlatos a készülékben hagyni. Kicserélésekor, ill. kivételekor a következő utasításokat tartsuk be!

Fontos!

1. Kapcsoljuk ki a készüléket! Az egység ki- és beszerelésekor az érintkezők véletlenül rövidre zárhatók; ezzel az egység (vagy a számológép) súlyosan megsérülhet, ha nem kapcsoltuk ki előzetesen a tápfeszültséget.
2. Húzzuk ki az ábra szerint a készülék hátulján levő modulfiók kis fedőlemezt! Ismét ügyeljünk arra, hogy testünk statikus töltését az egység érintése előtt levezessük.
3. Vegyük ki az egységet! A gépet ennél a műveletnél meg is fordíthatjuk, hogy az egység a kezünkbe essék.
4. Az egységet felíratával felfelé és rovátkolt részével befelé irányítva toljuk be a helyére. Az egységnek erőkiéjtés nélkül kell becsúsznia a megfelelő helyzetbe.
5. Töljük rá a fedőlemezt a fiókra, hogy az egység érintkezői biztosan érintkezzenek! Kerüljünk minden olyan tevékenységet, mely az érintkezőket elhajlítani, szennyezni vagy más módon megsérteni!



A software-programok lefuttatása

Ha behelyeztünk a készülékbe egy software-egységet, akkor annak egy meghatározott számú programja ugyanazzal a számmal hívható be. Egy gyűjtemény minden egyes programjának saját hívószáma van. A billentyűzés: 2nd Pgm mm, ahol mm a választott program kétjegyű sorszáma.

A programok felhasználásakor kövessük pontosan a programkézikönyv utasításait! Minden programhoz egy programcím-kártya tartozik. Ez a nemmágneses kártya megadja a programcím-billentyűkhöz rendelt feliratokat. A kártya a címkeívről való leválasztás után behelyezhető a billentyűk feletti ablakba.

Példaként számítsuk ki, mekkorára növekszik 1000 Ft 20 év alatt, ha a kamatláb 8%!

Ez a feladat az alapprogram-gyűjteményben levő kamatos kamat programmal könnyen kiszámítható.

Billentyű	Kijelzés	Megjegyzés
CLR 2nd Pgm 18	0	a gép beáll kamatos kamat programra (18. program)
2nd E'	0.00	beírás
20 A	20.00	az időtartam beírása
8 B	8.00	a kamatláb beírása
1000 C	1000.00	az alapösszeg (tőke) beírása
0 D	4660.96	a végösszeg kiszámítása kamatos kamattal.

A program a Pgm utasítás ismétlése nélkül bármikor lefuttatható, ha azt egyszer már kiválasztottuk. A programtárolóra az RST, a kézbillentyűzésre a 2nd Pgm billentyűvel lehet visszatérni, vagy áttérhetünk más programra. A software-programokat más programokba is be lehet hívni. (Lásd a IV. fejezetet az alprogramokról!) Ezzel a tulajdonsággal a gép programkapacitása jelentős mértékben növelhető. Ha bármikor tudni akarjuk, hogy melyik software-egység van a gépben, nyomjuk meg a 2nd Pgm 1 SBR 2nd Write billentyűket. Ekkor a gép kijelzi a modul számát és ha a gép nyomtatóhoz van kapcsolva, a fenti számot és a modul megjelölést is kinyomtatja.

A software-programok elemzése (programátvétel)

Valamennyi program közvetlenül hozzáférhető az egységen belül. Ilyen program alkalmazásakor a feldolgozási folyamat az egységbe közvetlenül lép be, és hajtja végre a feladatot. A software-program nemcsak köz-

vetlenül használható, hanem úgy is, hogy azt a gép programtárolójába bevisszük. Ily módon az egyes lépések elemezhetők, és egyéni szükségleteink szerint megváltoztathatók. Valójában a programnak csak egy másolata kerül át a programtárolóba, a változtatások ezen hajthatók végre, magának a software-egységnek az információtartalma nem változhat meg. A programátmásolás azonban egyszerű művelet.

1. Ellenőrizzük, hogy a programátvitelhez elegendő-e a programtárolási kapacitás. A tárolókapacitás elosztását a következő fejezet tárgyalja.
2. A 2nd Pgm billentyűzéssel meghatározzuk, hogy melyik programot akarjuk átvinni.
3. A 2nd Op 09 billentyűzéssel bemásoljuk a programot a programtárolóba.

Ez az eljárás a választott programot a 000 programhelytől kezdve viszi be a programtárolóba. Az átvett program átírja az ezen programtároló-részben levő korábbi utasításokat. Ebből kifolyólag a programtárolóba való software-programátviteli utasítás nem lehet része az egyéni programnak.

Mihelyt a programot átvittük a programtárolóba, céljainknak megfelelően megváltoztathatjuk. A módosított programot nem lehet visszavinni a software-egységbe.

Ha rögzíteni akarjuk az új programot, erre is van lehetőség. Az egyes lépéseket kódolt formában mágneskártyára, vagy -kártyákra lehet rögzíteni, vagy a nyomtatóval lehet kiírni. Ha egy software-egység védett programot tartalmaz, akkor ez nem másolható át. Ha utasítást adunk ilyen programok egyikének átadására, a kijelző villog. A software-modul hibáira nézve lásd az A. függelék.

IV. PROGRAMOZÁS

A PROGRAMOZÁS FOGALMA

A számítógép szerepe a mindennapi életben igen nagy, s így az olyan fogalmak mint programozó, programozás, vagy programnyelv, közismertek. Sokan ezeket a fogalmakat rendkívüli képességekkel hozzák összefüggésbe, s így a programozás megismerése is sok ember számára elérhetetlennek, vagy csak igen sok gyakorlással elsajátíthatónak tűnik.

Ez azonban nem így van, látni fogjuk, hogy a programkészítés egyszerű, némi gyakorlás után bárki elsajátíthatja. A PTK-1096 elég sokoldalú készülék ahhoz, hogy a programozás révén elérhető nagy számítási sebesség és teljesítmény a hasznossága mellett örömet okozzon, akár egyszerű aritmetikai feladatok megoldásáról, akár bonyolult műszaki számításokról legyen szó.

Egyszerűen kifejezve, egy program lényegében utasítássorozat, amelyet egy készülékkel vagy személyel közlünk. A számológép-program utasítja a gépet, hogyan dolgozza fel az adatokat, hogyan végezzen el egy számítást. Csak pontos utasításokat kell adni a gépnek, hogyan és mit kell tennie, és a gép megoldja a feladatot. A program pontos parancsok sora, melyeket meghatározott sorrendben, megbízhatóan végre kell hajtani.

A számológéppel való közlés eszköze a nyelv. Még a legegyszerűbb, négyműveletes számológépnek is van nyelve. A programozásra alkalmazva a nyelv az az eszköz, melynek segítségével a géppel programot közölhetünk.

A számológép nyelve a logika és a számtan nyelve. Ha tehát van tapasztalatunk a papíron ceruzával végzett számtani műveletekben, részben máris ismerjük a gép programnyelvét. A billentyűk működésére vonatkozó ismertetés is alkalmazható a programokban. A számológép a számítógéphez hasonlóan, csak a beadott parancsokat hajtja végre. A gépeknek ez a tulajdonsága különböző tapasztalatokhoz vezet a munka során. Ennek az a következménye, hogy a programozónak szigorúan ügyelnie kell arra, milyen utasításokat tárol a gép és az utasításokat milyen sorrendben kell beírni!

A számológép pontosan azt hajtja végre, amire utasítottuk, függetlenül attól, hogy mi a végrehajtást ilyen formában óhajtottuk-e vagy sem. Az itt részletezett módszerek bepillantást engednek a számológép nagy teljesítőképességébe és lehetővé teszik, hogy egyéni programot készítsünk.

A PROGRAMOZÁS ALAPJAI

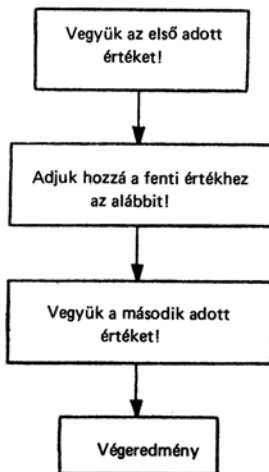
Változók bevétele a programba

Tekintsük a következő egyszerű kifejezést: $A+B=C$. Alapműveletekre alkalmas számológépnél az A és B értékeit nem lehet későbbi időpontban meghatározni. Azoknak már a bebillentyűzésekor ismerteknek kell lenniük. Ha egyszer egy kifejezést beírtunk és az eredményt megkaptuk, más változó esetén újra be kell írni a teljes kifejezést. A programozható számológépbe beírhatjuk az utasításokat anélkül, hogy az értékeket előre meghatároznánk, majd később már csak a változók adatait írjuk be és megkapjuk az eredményt.

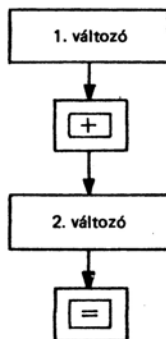
A fenti egyszerű számtani kifejezés esetén a négy alapműveletes és a programozható számológép számolási sebessége közel azonos. A programozható gép előnye az azonban már az alábbi kifejezés megoldásánál megmutatkozik. Tegyük fel, hogy az eredményt 10 különböző A -értékre akarjuk kiszámítani, B és C változatlanul hagyása mellett.

$$A \cdot \frac{B}{(1+A)^{-C}} =$$

Az egyenletet csak egyszer kell bebillentyűzni és utána az A értékét minden számításnál változtatni, ami a programozható számológépnek egyszerű feladat. Térjünk vissza ismét az első kifejezéshez: fontoljuk meg, hogyan lehet a gépnek utasításokat adni! Írjuk fel az utasításokat úgy, mintha azokat egy másik személy részére adnánk, majd az utasításokat alakítsuk át számítástechnikai nyelvre!



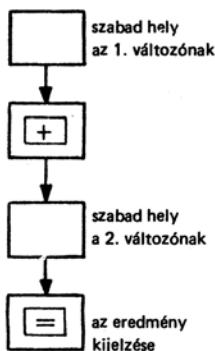
A és B tetszőleges értékek. Az ilyen értékeket változóknak nevezzük.



Ami a számológépet illeti, két feltételt kell kielégíteni ha a változókat nem a program részeként adjuk be.

1. az utasítássorozatban a megfelelő helyet üresen kell hagyni a változók későbbi beírása céljára.
2. meg kell adni a gép számára, hogy hol találhatók meg a változók értékei, ha szükség van rájuk. Ez az utasítás lehet az, hogy a változót a kijelzőből, vagy hogy az adattárolóból kell vennie.

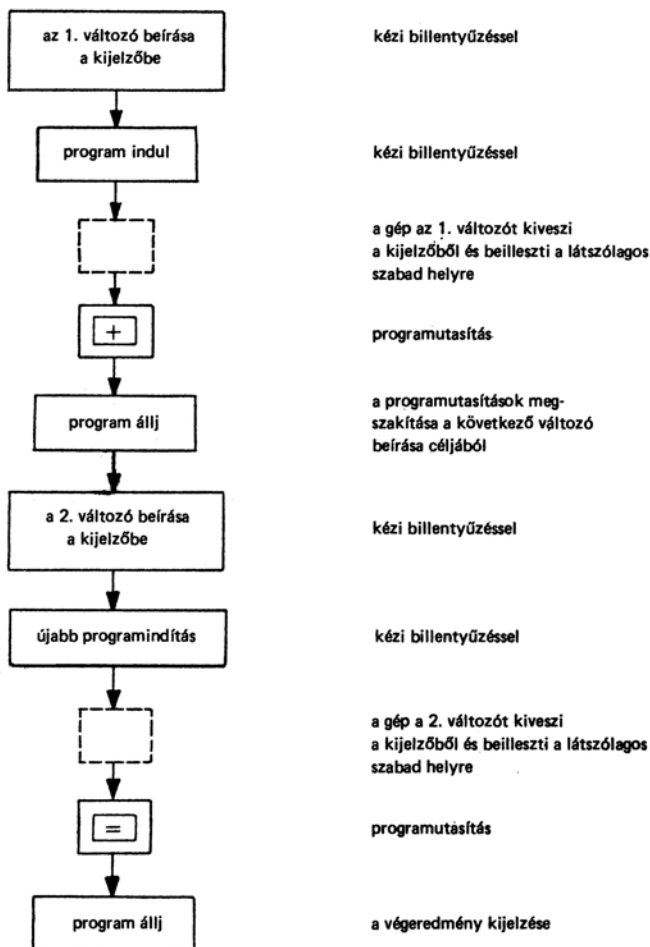
Végül is így alakul az utasítássor, figyelembe véve a változóknak szánt szabad helyet is:



Ezzel szabad helyet biztosítottunk, így a gép a kijelzett számot változóként használja. Ha a gép megkezdí a program lefuttatását, az éppen kijelzett érték bekerül az üres helyre. A maradék változót is figyelembe kell venni, tehát a programot meg kell szakítani, hogy a második változót is a kijelzőbe vihessük. Ha a program futtatása ismét elkezdődik, a gép felveszi a kijelzett értéket és folytatja a számítást.

Az egyik megoldás arra, hogy programunkban az új adatokhoz szabad helyet biztosítsunk az, hogy a szóban-forgó helyen a programot megszakítjuk az R/S billentyűvel (Run-Stop). Ily módon utasítjuk a készüléket arra, hogy minden információt tartson meg, miközben a második értéket a kijelzőbe billentyűzzük. Tulaj-donképpen látszólagos szabad helyről beszélhetünk, mivel a programutasítások közt a valóságban nincs ü. A programnak meghatározott helyen történő megszakításával jelezzük, hogy itt valamit végre kell hajtani, jelen esetben adatot kell bevinni.

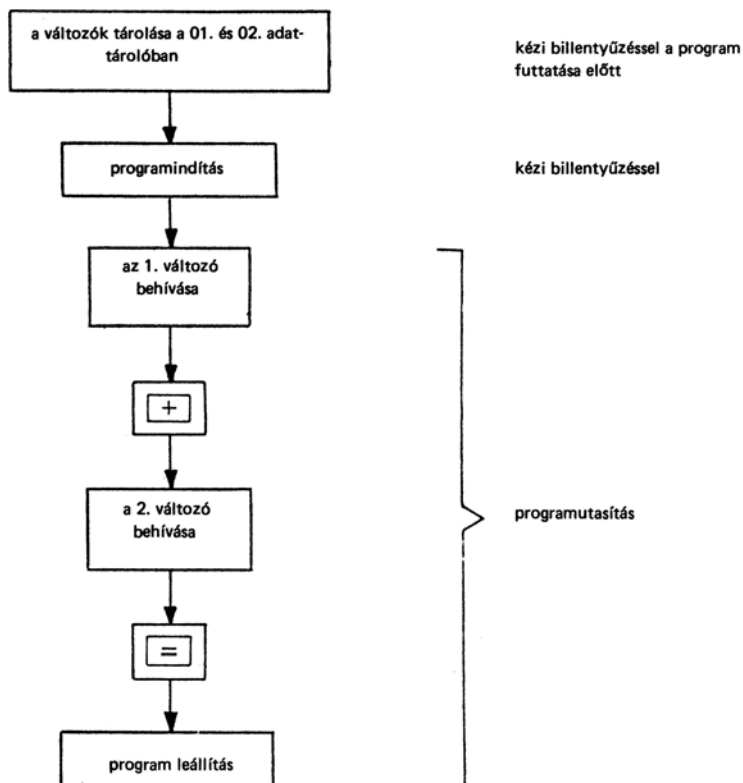
Folyamatábra, ha a változó a kijelzőben van



A fenti módszer, a programmegszakítás és adatbeírás abban az esetben ideális, ha minden programfuttatáshoz teljesen új változókat kell felhasználni. Előnyösebb lehet más eljárás alkalmazása, ha csak egy értéket kell megváltoztatni. Ennél az eljárásnál a gép adattárolóját használjuk fel a változók tárolására.

Ha a gépnek egy változót a tárolóban kell keresnie, olyan utasítást kell adnunk, amelynek segítségével a változót a megfelelő adattárolóból viszi be a programba. Pl. a 01 adattárolóban tárolt változót az RCL 01 billentyűzéssel hívjuk be.

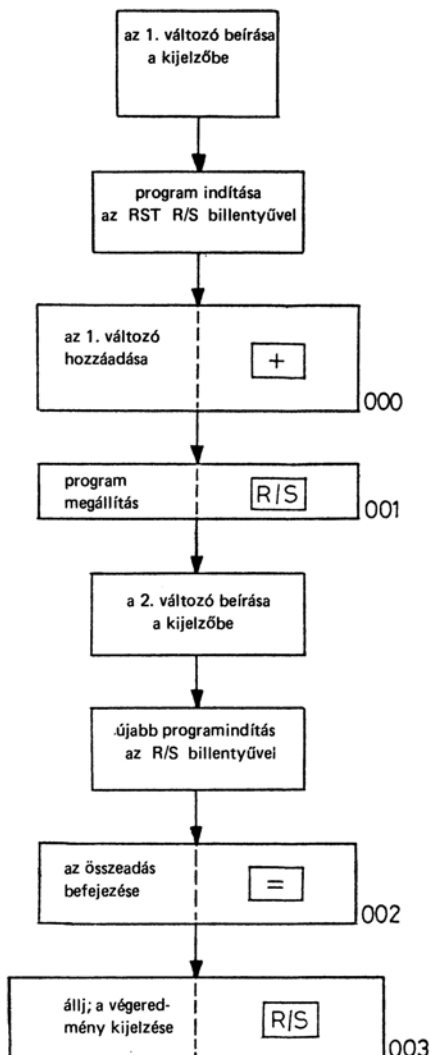
Folyamatábra, ha a változó az adattárolóban van



A következőkben röviden foglaljuk össze az eddigieket! Először megfogalmaztuk a feladatot. Ezt megmondások követték, a változó beírási módszereiről (tárolóval vagy enélkül.) Végül minden módszerhez egyszerű folyamatábrát készítettünk. Figyeljük meg, hogy a folyamatábra úgy keletkezett, hogy a feladatot lépésekre, ill. műveletekre bontottuk, amelyekkel a feladat megoldható, ha azokat felülről lefelé haladva végrehajtjuk.

A következő fontos lépés a folyamatábrának mint segédeszköznek az alkalmazása oly módon, hogy segítségével meghatározzuk azokat a billentyűzési utasításokat, amelyekre a gépnek a feladat megoldásához szüksége van. Ügyeljünk arra, hogy a következő példánál a program végrehajtása előtt az RST billentyűt kell benyomni! Ezáltal biztosítjuk, hogy a program valóban a 000 tárolóban levő utasítással kezdődjék, amikor a programot az R/S billentyűvel elindítjuk. A következő példa mutatja azoknak a billentyűknek a használatát, amelyekkel a gépet arra utasíthatjuk, hogy a változókat a kijelző tarolójából vegye.

Program, ha a változók a kijelzőben vannak



A folyamatábra tömbjei mutatják, hogy mit kell tennünk a program végrehajtása érdekében, miután az utasításokat a programtárolóba beírtuk. Ezeket az utasításokat, illetve a billentyűzést láthatjuk a szaggatott vonallal kettéválasztott tömbök jobb oldalán. A tömb jobb alsó sarkában kívül levő számok (pl. 000, 001) a tömbön részletezett utasítás számai. Ezeket a billentyű- vagy programutasításokat bevihetjük a programtárolóba, ha a gépet az ún. tanulási (learn) üzemmódba kapcsoljuk.

Az alábbiakban megadunk egy módszert a programok beadására.

1. Rövid időre kapcsoljuk ki a készüléket, majd újra kapcsoljuk be, a teljes törlés érdekében!
2. Kapcsoljuk a készüléket a LRN billentyűvel tanulási üzemmódba. Ez az üzemmód arról ismerhető fel, hogy a kijelzőn 000 00 jelenik meg.
3. Sorban nyomjuk be a folyamatábrán felsorolt billentyűket, fentről lefelé haladva. Ügyeljünk arra, hogy csak a feltüntetett billentyűket működtessük! Ha hibát követünk el, kezdjük újra az 1. lépésnél! A hibajavítás egyszerűbb módjáról később lesz szó.
4. A tanulási üzemmód leállítás céljából nyomjuk meg újból a LRN billentyűt! Ezáltal feloldjuk a különleges kijelzési formát is és csak egyetlen 0-t jelez ki a készülék. Most már végrehajthatjuk a programot.

A tanulási üzemmódban a kijelzőn látható három baloldali szám a program írása közben változik. Ezek a számok mutatják meg, hogy melyik programtárolóra irányítjuk a gépet, illetve hogy melyik parancsot kell legközelebbi lépésként végrehajtani. Tanulási üzemmódban a következő, el nem foglalt tárolóhelyszámot mutatja a kijelző.

Ezzel az ismeretekkel már megkísérelhetjük egy program végrehajtását abban az esetben, ha a változókat a kijelzőn keresztül adjuk be. A lépések:

1. bekapcsoljuk a számológépet
2. megnyomjuk a LRN billentyűt, így tanulási üzemmódba kapcsoljuk a gépet
3. beadjuk a programot az alábbi billentyűkkel

$$+ R/S = R/S$$

4. benyomjuk az LRN billentyűt, így feloldjuk a tanulási üzemmódot

Ezzel a gépet programoztuk. Most oldjuk meg a következő feladatot: $227+34=?$

1. A CLR billentyűvel töröljük az összes futó számítást.
2. Nyomjuk meg az RST billentyűt, és mint 1. változót, írjuk be a 227 értéket!
3. Nyomjuk meg az R/S billentyűt! A 227 a kijelzőben marad!
4. Írjuk be a 34 értéket, mint 2. változót, és ismét nyomjuk meg az R/S billentyűt! A kijelzőn megjelenő 261 az eredmény.

Téves eredmények elkerülésére ajánlatos a program végrehajtása előtt a CLR billentyűvel törölni a nem kívánt, függőben levő számításokat. Valahányszor megnyomjuk az R/S billentyűt, az értékek a kijelzőben összeadódnak, és a kijelző ezt ki is írja.

Ismételjük meg a programot más számértékekkel!

Most lássunk példát arra, amikor a változókat az adattárolóba helyezzük.

Írjunk ehhez egy új programot! (Lásd jobbra a folyamatábrát!)

Adjuk be a gépbe a programot az alábbiak szerint:

1. kapcsoljuk be a gépet,
2. az LRN billentyűvel kapcsoljuk a gépet programbeírási üzemmódba,
3. adjuk be a programot a következő billentyűzéssel:

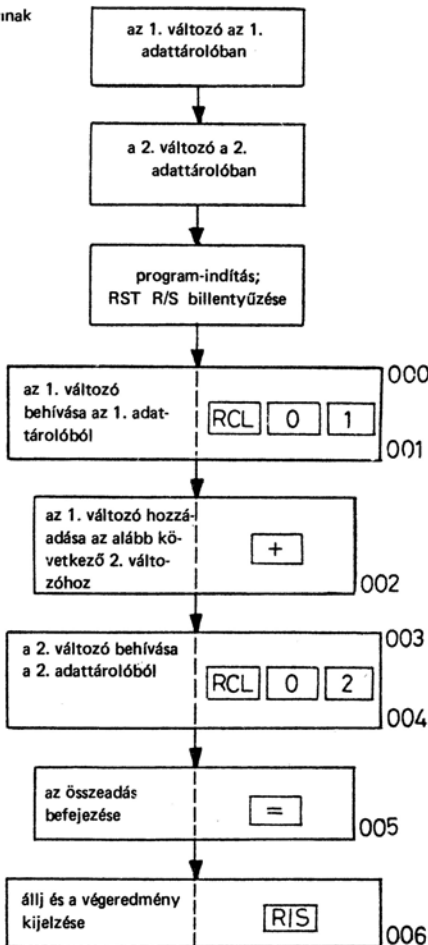
```
RCL
0
1
+
RCL
0
2
=
R/S
```

4. újra nyomjuk meg az LRN billentyűt!

Ez a program a változókat a 01 és a 02 jelű adattárolóból veszi. Tároljuk tehát a 227-t a 01 és a 34-t a 02 jelű adattárolóban!

- | | |
|------------------------|----------------------|
| 1. 227 az 1. tárolóban | 2. 34 a 2. tárolóban |
| a 227 beírása | a 34 beírása |
| STO 01 billentyűzése | STO 02 billentyűzése |

Program , ha a változók az adattárolóban vannak



Mint már a folyamatábrán jeleztük, a billentyűművelet a program végrehajtásához: RST R/S. Ha ezeket a billentyűket megnyomjuk, a kijelzőn megjelenik a 261, így a feladatot megoldottuk.

Figyeljük meg, hogy amikor a kijelzőbe billentyűztük be a változókat, kevesebb utasításra volt szükség! Következésképpen kevesebb programtárolóhelyet vettünk igénybe. Amikor változók tárolására adattárolót használunk, több utasítást kell adni a programtárolónak, de a gép az eredményt megszakítások nélkül számítja ki. A módszer kiválasztása az igényektől függ.

Gondoljuk végig a folyamatábra hasznosságát egy bizonyos feladat megoldásának megszervezése és ábrázolása szempontjából. A folyamatábra nemcsak az utasításokat és a programtárolóban levő programokat foglalja magába, hanem szemlélteti azokat a billentyűzéseket is, amelyek a program végrehajtásához szükségesek (pl. a programindítás és a változók beadása). Az ábrázolt billentyűzések azok az utasítások, melyeket a számítógép ismer és követ. Ezeket a tanulási üzemmódban a programtárolóban tároljuk: lényegében ez a program.

A programozás gyakorlata

Az aritmetika sokoldalú nyelve egyszerű, de igen összetett programok készítését is lehetővé teszi. Egyszerű programokat kevés fáradsággal számíthatunk, ellenőrizhetünk és végeztethetünk. Bár a nyelv közvetlen és a lehetőségekhez képest egyszerű, bonyolult programokat mégis gondosan meg kell tervezni. Ha még kevés programozási tapasztalatunk van, a következő javaslatok segítenek. Annak, aki már jártas a programozásban, ezek a javaslatok szolgáljanak ismeretei felfrissítésére és tájékoztatására! Tekintse az alábbi pontokat javaslatoknak, hiszen végül mindenki önálló programozási stílust alakít ki!

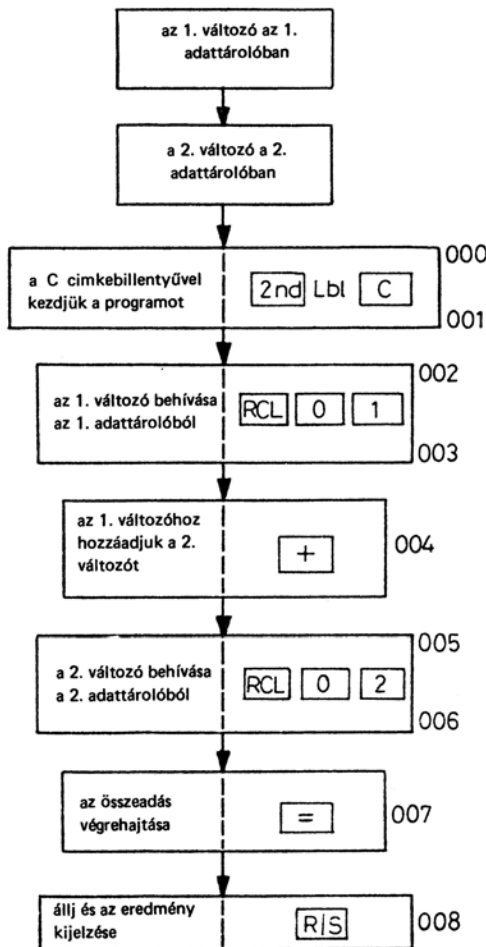
1. **Gondosan és világosan határozzuk meg a feladatot!** Határozzuk meg a képleteket, a változókat és a kívánt eredményt. Mi az ismert? Mit kell kiszámítani? Hogy viszonyulnak egymáshoz az ismert és ismeretlen értékek?
2. **A megoldási módszer (algoritmus) kialakítása.** A műveleti sorrend eldöntése a kívánt számérték meghatározásához, a rendelkezésre álló számítógép és programozási befogadóképesség figyelembevételével. Vegyük tekintetbe, hogy a feladatokat mi oldjuk meg és nem a gép! A gép csak utasításainkat fogja követni.
3. **A folyamatábra kialakítása.** Gyakran előnyös a feldolgozási folyamat ábrázolása vázlatok segítségével. Ily módon ábrázolhatjuk a különböző megoldási összetevők egymáshatását, és esetleg egyszerűsíthetjük a program szerkezetét.
4. **Kezdjük az adattároló hozzárendelésével!** Rendeljük hozzá a számokhoz, amelyekkel dolgozunk adattárolókat! Ezt a műveletet folytassuk az egész programozás alatt! Készítsünk jegyzetet arról, hogy melyik tárolóban milyen értéket tárolunk!
5. **A folyamatábra átírása billentyűzési sorrendre.** A kódformák megkönnyítik a munkát. Előnyös a címkek és tárolók kódolt alakú feljegyzése. Vezessünk egy megjegyzés rovatot a különböző programrészekre vonatkozó hivatkozások feljegyzésére!
6. **A program beadása.** Nyomjuk be a 2nd, a CP és a LRN billentyűt és billentyűzzük be a teljes kódolt programot! A beadás befejezéseként újra nyomjuk meg a LRN billentyűt a tanulási üzemmód leállítására céljából!
7. **A program felülvizsgálata.** Vizsgáljuk felül a programot, ellenőrző feladatok segítségével!
8. **Hibajavítás.** A program felülvizsgálatkor tapasztalt hibákat javítsuk ki a kódokon!
9. **A program átdolgozása.** Kapcsoljuk a készüléket tanulási üzemmódba, és hajtsuk végre a szükséges javításokat! Nyomjuk be ismét a LRN billentyűt és térjünk vissza a kézi üzemmódra!
10. **Ismételt programellenőrzés.** Szükség szerint ismételjük meg a 7., 8. és 9. lépést!
11. **A program felírása.** A gép mágneskártyás felírásra alkalmas, rögzítsük ezekre a programot!
12. **A programutasítások feljegyzése.** Ajánlatos a program használatával kapcsolatos utasításokat lépésről lépésre gondosan írásba foglalni. Még a legnagyobb teljesítményű programok is használhatatlanok, ha nem tudjuk, hogyan kell azokat végrehajtani. Rögzítsük részletesen azokat az információkat, amik a program végrehajtásához szükségesek!

A programcím-billentyűk alkalmazása (user-defined labels)

Az előző példák programjaiban ismételten használtuk az RST és az R/S billentyűket. Minthogy az RST a programszámlálót a 000 utasításra állítja vissza, ebből esetleg azt a következtetést vonhatja le valaki, hogy minden programnak a programtároló elején kell kezdődnie. Ha már több programozási tapasztalatunk lesz, látni fogjuk, hogy ez nem mindig célszerű. A számológépnek programcím-billentyűi vannak, ezekkel a címkekkel bármelyik tárolóhelyre közvetlenül rá lehet állni.

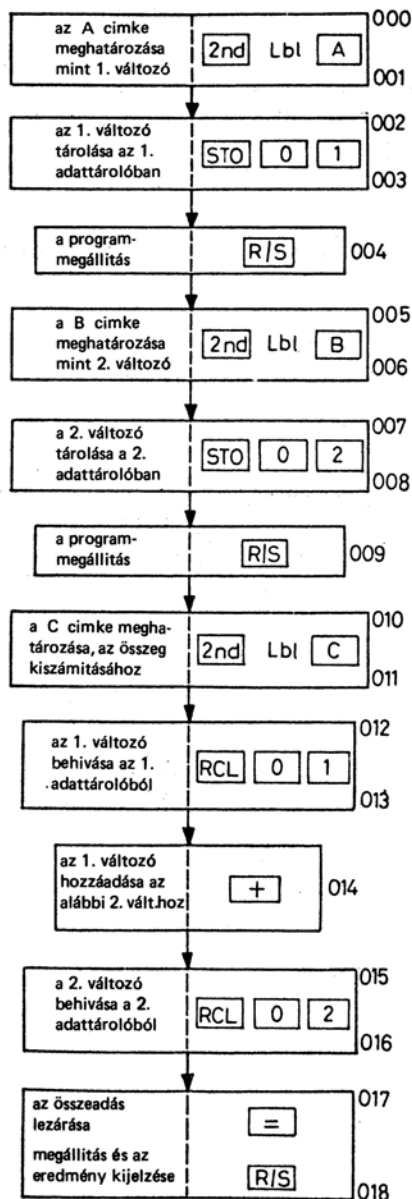
Ha programbillentyűt iktatunk a programba, a programszámláló a kérdéses címkére áll, mihelyt ezt a billentyűt megnyomjuk. A gép automatikusan úgy kezdi a program végrehajtását, hogy a számítások kiindulási pontja a címke utáni első utasítás legyen. Az A...E, valamint ezek másodlagos billentyűiről (2nd, A'...E') van szó. Ezekkel maximálisan 10 vonatkozási pontot lehet megjelölni a programok vagy részek programok számára. Ebben az esetben az RST R/S billentyűzés helyett a C vagy valamely másik prog-

ramcím-billentyűnek van szerepe; a 2nd Lbl címző billentyűk közbeiktatásával, amint ez a következő ábrából is látható.



Kapcsoljuk a gépet mint eddig, tanulási üzemmódban! Billentyűzzük be az ábrázolt programot! Ezután térjünk át alap-üzemmódba! Tároljuk a 227-t az 1. adattárolóban, valamint a 34-t a 2.-ban! Most nyomjuk be a C billentyűt, és megjelenik az eredmény. A C billentyűvel utasítjuk a gépet, hogy az Lbl C pozíciót keresse meg a programtárolóban, és az Lbl C szerinti utasításokat hajtsa végre.

A programcímzéssel kapcsolatos ismeretek elmélyítésére nézzünk egy újabb kiegészítési lehetőséget az előző programhoz! Lehetőség van arra, hogy az A billentyű megnyomásával az 1. változót az 1. adattárolóban, a B billentyű megnyomásával a 2. változót a 2. adattárolóban tároljuk. A C billentyűt ismét az eredmény kijelzésére használjuk fel.



Kapcsoljuk a készüléket tanulási üzemmódba, és billentyűzzük be a programot! Újból nyomjuk meg a LRN billentyűt a tanulási üzemmód feloldására! Vizsgáljuk meg a programot! Az 1. és a 2. változót tetszőleges sorrendben adhatjuk be.

Noha a program több lépésből áll, az alkalmazása sokkal egyszerűbb. A három program alábbi összehasonlítása képet ad arról, hogy a programcím-billentyű mennyire növeli a program használhatóságát. A címek a programba tetszőleges utasítási sorrendben vihetők be, a sor értelmének megváltozása nélkül.

Első változat	Második változat	Harmadik változat
a 227 beírása	a 227 beírása	a 227 beírása
STO 01	STO 01	A
a 34 beírása	a 34 beírása	a 34 beírása
STO 02	STO 02	B
RST R/S	C	C
kijelzés: 261	kijelzés: 261	kijelzés: 261

A címekre az utasítások feldolgozása során nem kell tekintettel lenni, szerepük csak az, hogy a program-tárolóban egy kívánt pontot meg tudjunk találni. A befejezetlen műveleteket nem befolyásolják. Ez nem jelenti azonban azt, hogy a címzés egy együtvé tartozó billentyűcsoportot (pl. RCL 14) megszakíthat: ezek egyetlen munkafázis meghatározására egynél több tárolóhelyet használnak fel. A címkeket már a program tervezésének elején figyelembe kell venni, és nem utólag bevezetni. A címkeket úgy kell billentyűzni a hátra-levő kódokkal együtt, mint a többi utasítást.

Természetesen egy olyan programot, mint két szám összeadása, a címzések nem tesznek egyszerűbbé, mert a billentyűzések száma nő, ahelyett hogy csökkenne. Mégis rá kellett mutatnunk, hogy nagyobb és bonyolultabb programoknál a címkek értékes segédeszközök.

Címzés rövidített formában

Eddig az adattároló egy helyének jelölésére kétjegyű címeket használtunk, azaz pl. az 1. adattárolóban tárolt változót az RCL 01 billentyűzéssel hívtuk be. Bizonyos esetekben a 0 ... 9 jelű adattárolókkal való műveleteknél az első nullák fölöslegesek. A címzésnek ez a módja az ún. rövidített alakú címzés, amire akkor van lehetőség, ha a tárolócímzés után közvetlenül nem számjegybillentyű következik.

Példaként tároljunk a 227-t az 1. adattárolóban, és 34-t a 2. tárolóban! Ezután behívjuk az értéküket és ki-számítjuk az összegüket.

Billentyű	Kijelzés	Megjegyzések
227 STO 01	227.	mivel utána számjegy-billentyű következik, a teljes címet meg kell adni
34 STO 2	34.	utóbbi 3 lépésnél rövidített alakú címzés
RCL 1 +	227.	alkalmazható, mert a cím után nem következik számjegy-billentyű
RCL 2 =	261.	

Figyeljünk meg, hogy rövidített alakú címzésnél az utasítás nem zárul le az utána következő nem-számjegy billentyű működtetéséig. Ez azt jelenti, hogy pl. a 227 nem hívódik be, amíg a + billentyűt meg nem nyomtuk; és a 34 sem tárolható, amíg RCL utasítást nem adtunk.

Egyéni program billentyűzése

A programozás olyan művelet, amely meghatározza a számológép számára az utasításokat. Ha az utasításokat már előkészítettük, alaposan meg kell ismerni a beírási eljárásokat is. Már ismerjük a tanulási üzemmódot, amelyet a következőkben részletesen tárgyalni fogunk.

A programot a feladat logikus átrendezésével állítjuk elő. A készülék a programutasításokat csak tanulási üzemmódban tudja elsajátítani. Megfordítva, a tanulási üzemmódban minden egyes billentyűzés utasítás a gép számára, kivételt képez a négy átrendező és hibajavító utasítás, ezeket néhány oldallal később részletezzük.

Ez a tény különösen fontos, mert azt jelenti, hogy az utasításokat gondosan, figyelmesen kell beírni. Prog-

ram üzemmódban billentyűzéssel nem végezhetünk számításokat.

Ha egy utasítás billentyűzése közben hibát követünk el, nem kell az egészet újra kezdenünk, és nem kell az utasításort újra beadnunk. A készülék hibajavító billentyűvel a hibás beírások helyesbíthetők, a hibás utasítások törölhetők, ill. a helyesek beiktathatók. Ezekkel a billentyűkkel kapcsolatban lásd A programok átdolgozása című fejezetet.

Ha megjegyezzük a következő egyszerű lépéseket, a program beírása nem okoz nehézséget.

1. A 2nd CP billentyűzéssel a programmutatót a 000 tárolóhelyre állítjuk vissza; továbbá a teljes programtárolót töröljük. Ez az utasítás feleslegessé teszi a készülék rövididejű ki- és bekapcsolását.
2. Kapcsoljuk a készüléket tanulási üzemmódba az LRN billentyűvel! A kijelzési alak magyarázatát A program kijelzése című fejezetben találjuk.
3. Billentyűzzük be a programot és ne feledkezzünk meg a 2nd billentyűről!
4. Ügyeljünk arra, hogy program ne lépje túl a programtároló befogadóképességét! Ha túl sok utasítást adunk be, a gép kézi billentyűzésre kapcsol és a tanulási üzemmódra jellemző kijelzés megszűnik.
5. Újra nyomjuk be az LRN billentyűt és ezzel álljunk vissza kézi billentyűzésre!
6. Hajtsuk végre az ellenőrző programot, és szükség esetén dolgozzuk át vagy helyesbítsük! Lásd a Programok átdolgozása című fejezetet.

A program kijelzése

Tanulási üzemmódban a kijelzőnek mutatnia kell a programszámláló állását, valamint az ezen tárolási helyen levő pillanatnyi utasítást. Kapcsoljuk be a készüléket és állítsuk a LRN billentyűvel tanulási üzemmódba! Ekkor a kijelzőn két, nullákból álló csoportnak kell megjelennie.

000 00

A hármas csoport baloldalt a tárolási hely számát mutatja, melyre a programtárolót vezéreltük. A program beírásakor minden egyes utasítás helyét kijelöljük a programtárolóban. Ezáltal nemcsak állandóan ellenőrizhetjük az utasításokat, hanem meghatározzuk a gép számára, hogy az utasításokat milyen sorrendben hajtsa végre. A programtároló általános esetben 480 programlépést tárolhat (lásd az V. fejezet Tárolókapacitás megválasztása c. részt).

Mint hogy a gép csak számokat tud feldolgozni, minden billentyűhöz egy kétjegyű kódot rendelünk hozzá (billentyűkód). A kijelzőben levő jobboldali kétjegyű szám mutatja annak az utasításnak a kódszámát, amely a baloldalt egyidejűleg kijelzett programtárolási helyen van. A billentyűk kódolására szolgáló általános szabályok a következők:

1. A számjegy-billentyűk kódszámát a megfelelő számjeggyel képezzük, így pl. a 7 billentyűt 07-ként kódoljuk.
2. Minden egyéb billentyű alaprendeltetésének kódszáma a billentyűzeten elfoglalt helyzetéhez igazodik. Az első számjegy (1 ... 9) megmutatja, hogy a kilenc sor közül, fentről lefelé haladva melyikben található a billentyű. A második számjegy (1 ... 5) azt mutatja, hogy a soron belül hanyadik oszlopban található balról jobbra. Pl. a tanulási üzemmódban a $\sqrt{x}$ utasítás a 3. sor 4. oszlopában 34-ként van jelölve, és ha pl. a 073 programtárolóhelyet foglalja el, a kijelzés a következő:

073 34

3. Másodlagos rendeltetést úgy kódolunk, hogy az oszlopszámhoz mindig ötöt hozzáadunk. Így pl. a $\sqrt{x}$ billentyű másodlagos rendeltetése a cos függvény. 2nd cos billentyűzés esetén a gép 34 helyett 39-et kódol. A 2nd tan billentyűzéshez viszont nem 40 (35+5), hanem 30 tartozik, mert a sor száma nem változott.

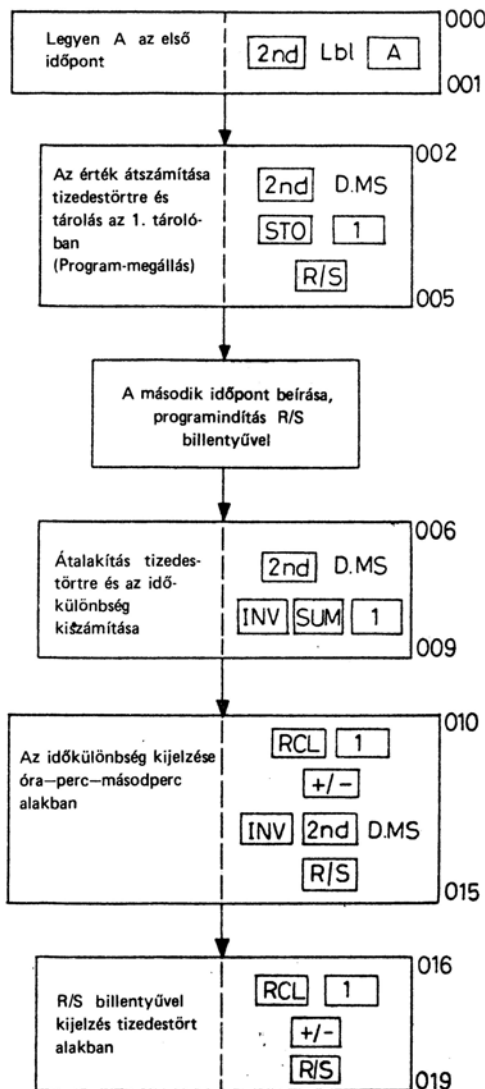
A billentyűkódok jegyzéke az V. fejezetben található.

A kódok megismerése céljából, a tanulás megkönnyítésére a készülékre tehetjük a tartozékként szállított át-látszó fóliára nyomott kódjegyeket. Néhány különleges esetben az utasítások kombinációja jelenhet meg egyetlen tárolóhelyen. Ezekben az esetekben a billentyűkódok is kombinálódnak és egyetlen kétjegyű kód-dá olvadnak össze. Az RCL 12, pl. két helyet foglal el a programtárolóban. Ekkor az RCL kerül az első helyre, az 1 és a 2 viszont összevonva a második tárolási helyre jut. Ez esetben a 12 nem a B program-

címutasítást jelenti, mert minden tárolási utasítás után a gép a következő utasítást egy adattároló címmé alakítja. Az ettől eltérő különleges eseteket az adott alkalommal tárgyaljuk.

Időkülönbség-program

Írjunk programot két adott időpont közötti időkülönbség kiszámítására! Az időt írhatjuk óra-perc-másodperc formában is (azaz $3^h, 16' 03'' = 3.1603$), és a számítás céljára a 2nd D.MS billentyűzéssel alakíthatjuk át tizedestörtre. Részleteket ezzel kapcsolatban a Régi fok átszámítása c. alfejezet tartalmaz a II. fejezetben.



Végezzük el ezt a feladatot a következők szerint!

Először nyomjuk be a 2nd és a CP billentyűket, hogy a programtárolót töröljük, és a programmutatót a 000 tárolóhelyre állítsuk! Ezután írjuk be a programot az alábbiak szerint!

Billentyű	Kijelzés	Billentyű	Kijelzés
LRN	000 00	RCL	011 00
2nd Lbl	001 00	1	011 00*
A	002 00	+/-	013 00
2nd D.MS	003 00	INV	014 00
STO	004 00	2nd D.MS	015 00
1	004 00*	R/S	016 00
R/S	006 00	RCL	017 00
2nd D.MS	007 00	1	017 00*
INV	008 00	+/-	019 00
SUM	009 00	R/S	020 00
1	009 00*	LRN	0

* Ezek a kijelzések nem látszanak következetesnek, holott arról van szó, hogy itt az ún. rövidített alakú címzést alkalmaztuk. A gép tehát az adattárolócímzés befejezését várja. Az első esetben pl. 005 00 jelenne meg a kijelzőben, ha az STO 01 billentyűket nyomnánk meg. Mivel azonban a fenti példában a nem számjegyes R/S utasítás következik, így tudja meg a gép, hogy az előző lépéssel a címzés befejeződött és azt 005 helyre kell tárolnia. A kijelző automatikusan 006-ra lép tovább, mivel a 005 hely betelt.

Mint ahogy kezdetben a programtárolót a 2nd CP billentyűzéssel töröltük, a program beadásakor minden billentyűkód 00 lesz. Ennek az az oka, hogy a programmutató egy tárolóhely elfoglalása után azonnal tovább lép a következő helyre és a kijelző nem az éppen beadott parancs kódját, hanem a következő helyen tárolt utasítás billentyűkódját jelzi ki.

A következő módszerrel megvizsgálhatjuk, hogy a programot helyesen adtuk-e be. Ha van KA-100 típusú nyomtatónk, akkor a programsor kinyomtatására nyomjuk be a következő billentyűket: RST 2nd List!

A sornyomtató a programlépés számát, a billentyű kódját és a billentyű feliratát írja ki. A kinyomtatást az R/S billentyűvel lehet leállítani.

Az egyes programlépések az alábbi módon jelezhetők ki a számológéppel.

Billentyű	Kijelzés	Utasítás
RST LRN	000 76	2nd Lbl
SST	001 11	A
SST	002 88	2nd D.MS
SST	003 42	STO
SST	004 01	1
SST	005 91	R/S
SST	006 88	2nd D.MS
SST	007 22	INV
SST	008 44	SUM
SST	009 01	1
SST	010 43	RCL
SST	011 01	1
SST	012 94	+/-
SST	013 22	INV
SST	014 88	2nd D.MS
SST	015 91	R/S

SST	016 43	RCL
SST	017 01	1
SST	018 94	+/-
SST	019 91	R/S
LRN	0	

Mivel a programozás hibátlan, futtassuk le a programot a következő értékekkel: 2h 15' és 3h 42' 54"

Beírás	Billentő	Kijelzés	Megjegyzések
2.15	A	2.25	t_1 (H.MMSS) $\rightarrow t_1$ (H.hh)
3.4254	R/S	1.2754	t_2 (H.MMSS) $\rightarrow \Delta t$ (H.MMSS)
	R/S	1.465	$\rightarrow \Delta t$ (H.hh)

Az időkülönbség 1 óra 27 perc és 54 másodperc, illetve 1.465 óra. Ha a fenti program lefuttatásakor a kiírt mennyiség óra-perc-másodperc alakú és értéke pl. 6.396, ez 6 órát, 39 percet és 60 másodpercet, ill. 6 órát és 40 percet jelent.

A programok átdolgozása

Tanulási üzemmódban a következő beavatkozási lehetőségeink vannak:

1. a tárolt utasítás kijelzése a tárolási hely megadása alapján
2. egy utasítás helyettesítése egy másikkal,
3. egy utasítás törlése és a folytonossági hiány megszüntetése,
4. helybiztosítás egy utólagos utasításhoz, az előzőleg programozott utasítások megzavarása nélkül,
5. lépésként való áthaladás a programtárolón előre vagy hátra; anélkül, hogy annak tartalmát befolyásolnánk.

Ezek a tulajdonságok lehetővé teszik a program ellenőrzését, javítását és módosítását anélkül, hogy a helyes utasításokat újból be kellene írni. Azok a billentyűk, amelyek megnyomása tanulási üzemben sem jelent programutasítást az SST, BST, INS és Del. Az SST és a BST billentyűvel lépésként áthaladhatunk a programtárolón, előre vagy hátra, és megnézhetjük az egyes tárolóhelyek programutasítását. Számítási üzemmódban az SST billentyűvel a program lépésként hajtható végre, ami lehetőséget ad az egyes műveletek eredményeinek megfigyelésére. A 2nd INS utasítás hatására a programtárolóban a pillanatnyi utasítás és az összes következő utasítás egy hellyel hátrább lép, és a pillanatnyi tárolóhelyre a gép nulla-műveletet iktat be. Ha a programtároló utolsó helyén volt utasítás, akkor ez az INS utasítás révén törölődik. A 2nd Del billentyűzéssel a programtárolóhelyen pillanatnyilag levő utasítást töröljük, és az összes következő utasítás egy tárolási hellyel előbbre lép. Így módon az utolsó tárolási helyre egy nulla-művelet kerül.

Két további célszerű szerkesztő billentyű az RST és a GTO (Vigyázat! Az RST-nek több rendeltetése van; lásd az Alapvető programvezérlési utasítások című részt az V. fejezetben.) Ha az RST utasítást kézzel billentyűzve adjuk be, akkor a programmutató 000 értékre áll be. A GTO utasítás, ha azt egy háromjegyű abszolút programcím vagy egy címke követi, a programmutatót erre a programtárolási helyre viszi. Az első helyen álló nullák rövidített alakú címzésnél elhagyhatók. Ha megnyomjuk a GTO billentyűt és utána címke-címzés következik, akkor a programmutató a címke utáni első tárolóhelyre áll be. Ha a fenti billentyűzések valamelyike után kapcsolunk tanulási üzemmódba, akkor a kívánt tárolóhelyen megvizsgálhatjuk a programtároló tartalmát. Vigyázat! Ha ezeket a billentyűket olyankor nyomjuk be, mikor a készülék tanulási üzemmódban van, akkor a gép azokat programutasításként érzékeli.

A billentyűk alkalmazását részletesebben az V. fejezetben ismertetjük (A programok módosítása).

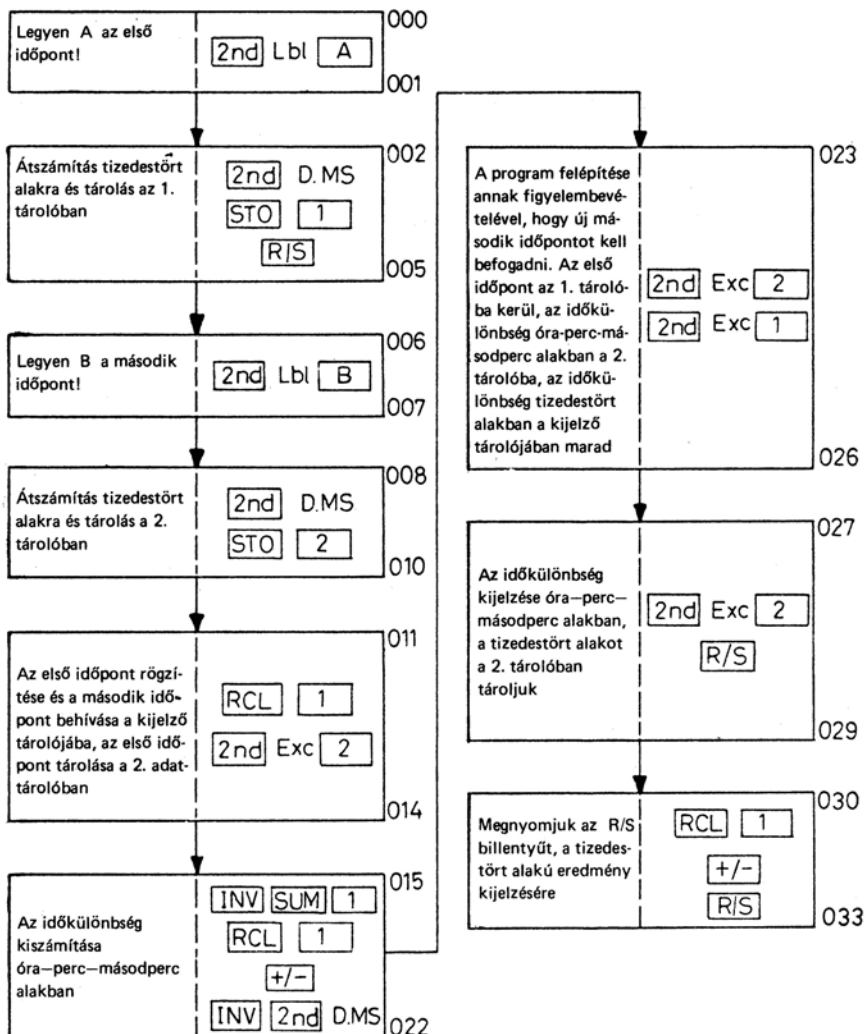
Ha tehát valahol át akarjuk írni a programot, keressük meg a helyet a fent leírt módszerek egyikével, és egyszerűen írjuk át a régi utasításokat újakra, tegyünk hozzá, vagy töröljünk szükség szerint!

Az időkülönbség-program javítása

Alakítsuk át a legutóbbi programot úgy, hogy a második időpontot az első időpont beadásának megismétlése nélkül meg lehessen változtatni. Fogjunk hozzá a változtatáshoz az eredeti program beadása után anélkül, hogy újra billentyűznénk az egész programot!

Három követelményt kell kielégíteni. Először is a második időpont beadásához egy címkét kell hozzárendel-

ni, hogy ehhez a programszakaszhoz közvetlenül hozzá lehessen férni. Másodszor szükségünk van egy olyan eszközre, mellyel az első időpontot meg tudjuk tartani, vagyis, hogy ez az információ a számítás után is hozzáférhető legyen. Harmadszor pedig a programot úgy kell felépíteni, hogy a számítás elvégzése után a második időpont új értékét be tudja fogadni.



A fent megadott billentyűzésbe hiba csúszott. Kísérlelje meg a hiba felderítését, mielőbb tovább futtatnánk a programot! Hajtsuk végre a szükséges változtatásokat a következő módszer szerint!

Billentyű	Kijelzés	Megjegyzés
GTO 06	0.	A programmutató beállítása a 006 tárolóhelyre
LRN	006 88	A tanulási üzemmód beírása
2nd Ins	006 00	A B címke beiktatása
2nd Ins	006 00	
2nd Lbl	007 00	
B	008 88	
SST	009 22	A programmutató előreléptetése egy lépéssel
2nd Ins	009 00	A 009..014 utasítások beiktatása
2nd Ins	009 00	
STO 2	010 00	
2nd Ins	011 00	
2nd Ins	011 00	
RCL 1	012 00	
2nd Ins	013 00	
2nd Ins	013 00	
2nd Exc 2	014 00	
SST SST SST	018 43	A programmutató előreléptetése 8 lépéssel
SST SST SST	021 22	
SST SST	023 91	
2nd Ins	023 00	A 023..028 utasítások beiktatása
2nd Ins	023 00	
2nd Exc 2	024 00	
2nd Ins	025 00	
2nd Ins	025 00	
2nd Exc 1	026 00	
2nd Ins	027 00	
2nd Ins	027 00	
2nd Exc 2	028 00	
LRN	0.	A tanulási üzemmód törlése

A következő módszerrel ellenőrizhetjük, hogy a programot helyesen változtattuk-e meg.

Billentyű	Kijelzés	Utasítás
RST LRN	000 76	2nd Lbl
SST	001 11	A
SST	002 88	2nd D.MS
SST	003 42	STO
SST	004 01	1
SST	005 91	R/S
SST	006 76	2nd Lbl
SST	007 12	B
SST	008 88	2nd D.MS
SST	009 42	STO
SST	010 02	2
SST	011 43	RCL
SST	012 01	1
SST	013 48	2nd Exc
SST	014 02	2
SST	015 22	INV
SST	016 44	SUM
SST	017 01	1
SST	018 43	RCL

SST	019 01	1
SST	020 94	+/-
SST	021 22	INV
SST	022 88	2nd D.MS
SST	023 48	2nd Exc
SST	024 02	2
SST	025 48	2nd Exc
SST	026 01	1
SST	027 48	2nd Exc
SST	028 02	2
SST	029 91	R/S
SST	030 43	RCL
SST	031 01	1
SST	032 94	+/-
SST	033 91	R/S
LRN	0.	

Hajtsuk végre a programot a következő értékekkel! Az első időpont 2:13:57, a második időpont 2:14:24.

Beírás	Billentyű	Kijelzés	Megjegyzés
1.3	A	1.5	t_1 (HH.MMSS) $\rightarrow t_1$ (HH.hh)
2.1357	B	0.4357	t_2 (HH.MMSS) $\rightarrow \Delta t$ (HH.MMSS)
	R/S	-1.5	?

A példát itt félbeszakítjuk, mert az eredmény nyilván hibás. A kijelzett mennyiségnek időkülönbségnek kellene lennie decimális alakban. Itt a gép az első időpont negatív értékét jelzi ki decimális alakban. A folyamatomatára és a hozzárendelt billentyűutasítások felülvizsgálata mutatja, hogy a kívánt eredmény nem vész el. Csak az történt, hogy a 023 ..028 cserebillentyűzés az információt a 2. adattárolóba vitte át. Tehát a hiba megszüntethető, ha a 31. lépést megváltoztatjuk. A javítás egyszerűen az új utasítás beírásából áll.

Billentyű	Kijelzés
GTO 3 1	-1.5
LRN	031 01
2	032 94
LRN	-1.5

Most futtassuk le ismét a programot!

Beírás	Billentyű	Kijelzés	Megjegyzés
1.3	A	1.5	t_1 (HH.MMSS) $\rightarrow t_1$ (HH.hh)
2.1357	B	0.4357	t_2 (HH.MMSS) $\rightarrow \Delta t$ (HH.MMSS)
	R/S	0.7325	$\rightarrow \Delta t$ (HH.hh)
2.1424	B	0.4424	t_2 (HH.MMSS) $\rightarrow \Delta t$ (HH.MMSS)
	R/S	0.74	$\rightarrow \Delta t$ (HH.hh)

Programátírás kombinált kódokkal

Tegyük fel, hogy a kijelzendő információ a fenti példában nem a 2., hanem a 12. tárolóban van. Ebben az esetben a 031 tárolóhelyre a 12 számot kellett volna tárolni. Itt különösen ügyelni kell, hogy ezt a kódot helyesen kombináljuk. Lásd a Program kijelzése c. fejezetet néhány oldallal előbb! Most lássunk egy módszert ennek a kódnak beírására! Ennél a módszernél az RCL utasítást újból be kell írni. Ezzel arra utasítjuk a gépet, hogy az adattároló címét automatikusan összefoglalja, és egyetlen helyre tárolja.

Billentyű	Kijelzés
CLR GTO 30	0.
LRN	030 43
RCL	031 02
1 2	032 94
LRN	0.

Mielőtt tovább haladnánk, fontosnak tartjuk megjegyezni, hogy a programozás egyéni jellegű tevékenység. Ez azt jelenti, hogy két különböző személy, akik ugyanazt a programot készítik, különböző utasításokkal juthatnak ugyanarra az eredményre. Már a feladat-megoldások különbözőségén is felismerhetők az egyéni vonások. A szervezési megoldások a képzettségtől és a szakmai tapasztalatoktól függően különbözhetnek. Nagy matematikai gyakorlattal rendelkező mérnök valószínűleg összetett matematikai egyenletekkel jut a megoldáshoz, míg a szellemtudományok embere a feladatot — kevés matematikai gyakorlata lévén — egyszerű számtannal oldja meg. Az egyik személy megelőlegszik olyan utasítássorozattal, mely a programtároló nagy részét lefoglalja, míg a másik állandóan keresi a programsűrítés lehetőségeit, hogy minél kisebb tárolási kapacitást vegyen igénybe. Mindegyikük megtalálja azt a megoldási módszert, melyben a legáratosabb. Példa: 1000,— Ft összeg évi 5%-os kamattal az év végére 50 Ft-tal nővekszik 1050 Ft-értékre. Az 1000 Ft a kezdőtőke, ez nem tartalmaz kamatot. Az év végén az 1050 Ft-ra megnövekedett értéket végösszegnek nevezzük. A kamatos kamatszámítás azt jelenti, hogy egy meghatározott pénzösszeget befizetünk egy számlára és több periódusidőn át ott hagyjuk. A kamatot a periódusok kezdetén mindig hozzáadjuk a kiindulási pénzösszeghez. Így módon a következő periódusban a kamatos kamatot kamatoztatjuk. Az eredeti betétösszeg a következőképpen módosul:

$$1000 + 1000 \cdot (0,05) = 1050 \text{ az első év végén,} \\ 1050 + 1050 \cdot (0,05) = 1102,50 \text{ a második év végén.}$$

Ennek a folyamatnak képlete általánosan ismert.

Matematikailag

$$y = x \cdot \left(1 + \frac{p}{100}\right)^n$$

Mielőtt a programot leírjuk, logikusan tervezzük meg, hogy mit kell tenni! Először a kamatlábat kell az egyenletbe beírni százalékos alakban, ezt a gép a beírás után 100-zal elosztja. A takarékpénztárak viszont különböző kamatozási periódusokat jelölnek meg (negyedév, nap stb.). A program sokoldalúvá válik, ha a készüléssel közöljük a kamatozás módját. A változásokat bevezetve a végösszeg számítására szolgáló egyenlet a következőképpen alakítható át

$$y = x \cdot \left[1 + \left(\frac{p}{100 \cdot c}\right)\right]^{c \cdot n}$$

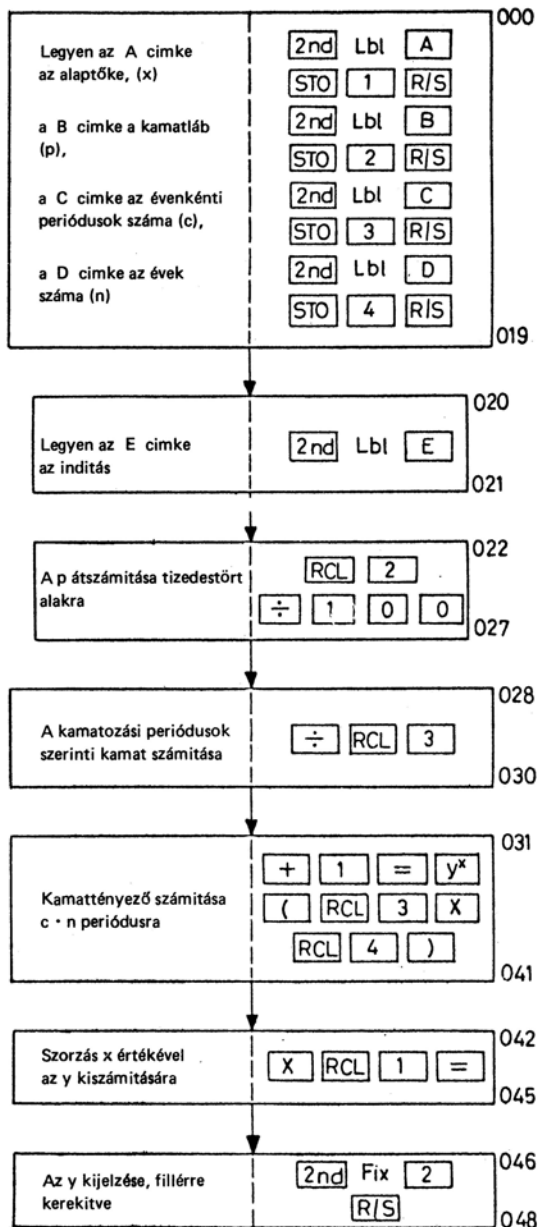
A képletben alkalmazott változók:

- y — a betét végértéke,
- x — a betét kiindulási értéke
- p — az éves kamatláb
- c — az évenkénti jóváírások száma
- n — a teljes kamatozási idő években.

Megjegyzés: Európában a kamatos kamatszámításhoz általában a tényleges évi kamatot használják. A tényleges évi kamatok nem egyszerű többszörösei a jóváírási időszakra vonatkozó kamatoknak, hanem maguk is kamatos kamatszámítással jönnek létre. Az egyszerűség kedvéért itt lemondunk az európai módszerről.

El kell döntenünk, hogy a változókat a kijelzőbe írjuk be, amikor azokra szükség van, vagy tároljuk az értékeket, hogy szükség esetén behívhassa azokat! Ebben a példában tároltuk a változókat. Így a változókat egyenként lehet beírni és a különböző eseteteket könnyebben lehet számolni. Ha újra végrehajtjuk a programot a korábban beírt adatokkal, akkor gondoskodnunk kell arról, hogy az eredeti adatokat megtartsuk. Hasonló változtatásra volt szükség az időkülönbséget számító programban.

A megoldási módot tehát rögzítettük, az egyenlet felépítésében tartalmazza a kívánt változókat, sőt a változók feldolgozásának módját is meghatároztuk. Most már kialakíthatjuk a folyamatábrát és leírhatjuk a programot.



PROGRAMUTASÍTÁSOK

Lépés	Eljárás	Beírás	Billentyű	Kijelzés
1.	A programtároló törlése és a programmutató visszaállítása		2nd CP	0.
2.	Tanulási üzemmód		LRN	000 00
3.	Kamatos kamatprogram beírása		LRN	0
4.	A tanulási üzemmód feloldása		A	x
5.	Az alapösszeg beírása	x	B	p
6.	Az évi kamatláb beírása	p	C	c
7.	Az évi jóváírási periódusok beírása	c	D	n
8.	Az évek számának beírása	n	E	y
9.	A végösszeg kiszámítása A változók beírásának sorrendje tetszőleges, új feladatok esetén a változatlan értékű változókat nem kell újra beírni.			

Tárolási hely és

billentyűkód

Billentyűzés

000 76	2nd Lbl
001 11	A
002 42	STO
003 01	1
004 91	R/S
005 76	2nd Lbl
006 12	B
007 42	STO
008 02	2
009 91	R/S
010 76	2nd Lbl
011 13	C
012 42	STO
013 03	3
014 91	R/S
015 76	2nd Lbl
016 14	D
017 42	STO
018 04	4
019 91	R/S
020 76	2nd Lbl
021 15	E
022 43	RCL
023 02	2
024 55	÷

Tárolási hely és

billentyűkód

Billentyűzés

025 01	1
026 00	0
027 00	0
028 55	÷
029 43	RCL
030 03	3
031 85	+
032 01	1
033 95	=
034 45	y ^x
035 53	(
036 43	RCL
037 03	3
038 65	X
039 43	RCL
040 04	4
041 54	)
042 65	X
043 43	RCL
044 01	1
045 95	=
046 58	2nd Fix
047 02	2
048 91	R/S

Számítsuk ki a 3000 Ft értékű betét végösszegét 5 év múlva, ha az évi kamat 8%, és naponta ill. havonta történik a jóváírás.

Beírás	Billentvény	Kijelzés	Megjegyzések
3000	A	3000.	x
8	B	8.	p
365	C	365.	c
5	D	5.	n
	E	4475.28	y
12	C	12.00	c
	E	4469.54	y

Árszámítási program

Eddig az adattárolót elsősorban változók tárolására és behívására használtuk. A számológép a tárolóban előzőleg tárolt változókkal, azok behívása nélkül is el tudja végezni az összeadás, kivonás, szorzás és osztás műveletét. A tárolónak ilyen módon való felhasználását tároló aritmetikának is nevezzük.

A tároló aritmetika előnyét a következő mintaprogramon mutatjuk be. Figyeljük meg, hogy egyszerű aritmetikával milyen célzerű programot lehet kialakítani! Nyomatékosan hangsúlyozzuk, hogy a programozható számológépek mindenféle programhoz előnyösen használhatók, nemcsak bonyolult matematikai feladatok megoldására.

Egy vállalatnál levő megrendelések különböző értékű, különféle árucikkból tevődnek össze. A vevő számláját úgy állítják ki, hogy az egységárat megszorozzák az árucikk megrendelt darabszámával, és így megkapjuk a számbanforgó cikk szállított mennyiségének árát. Egy rendelés átlagos egységárának kiszámításához összeadjuk a rendelt mennyiségeket, és ezzel az összeg $\sum_{i=1}^n$ elosztjuk a számlán feltüntetett összeget. Ez nem nehéz, de időrabló művelet.

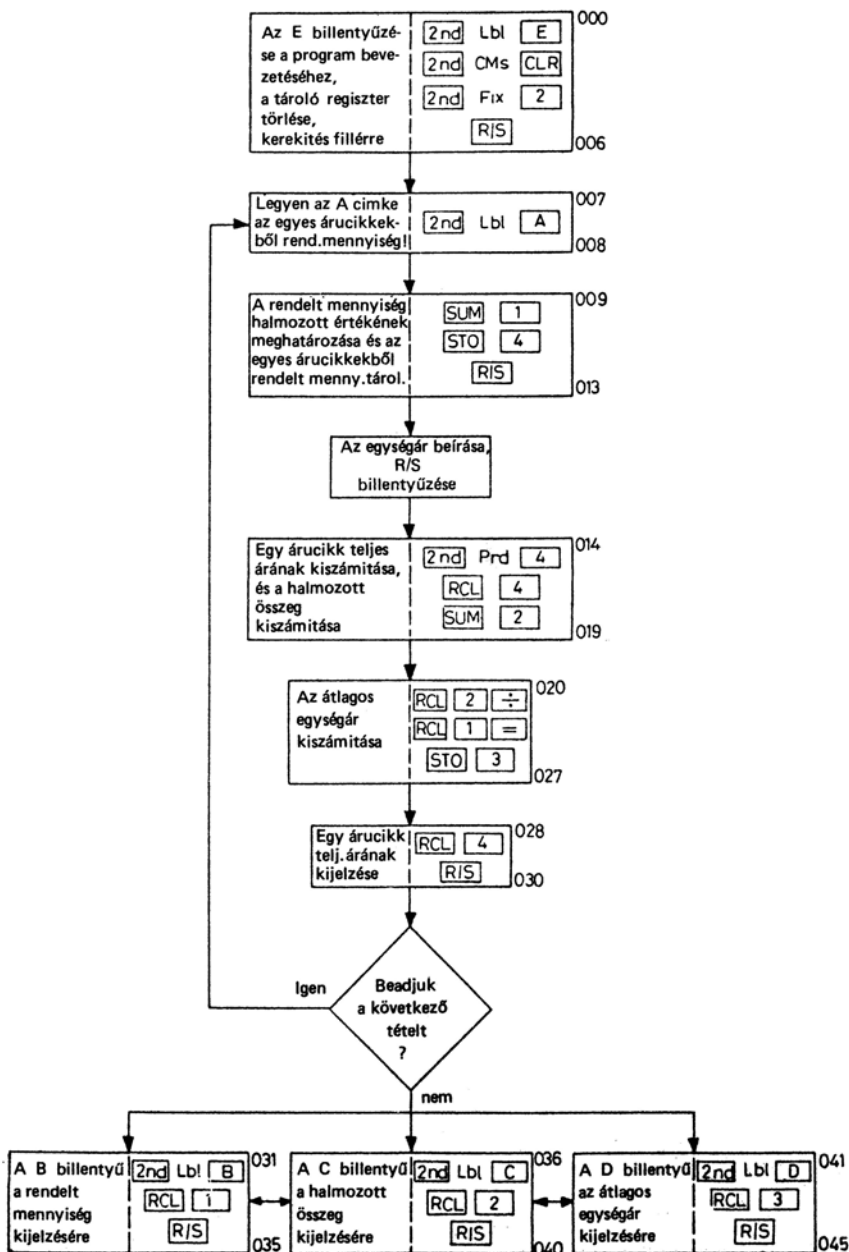
Árucikk	Rendelt mennyiség	Egységár (Ft)	Teljes ár (Ft)
1	100	0,25	25,00
2	200	0,15	30,00
3	50	0,35	17,50
4	150	0,40	60,00
5	300	0,10	30,00
Teljes rendelt mennyiség	800		162,50
A rendelés átlagos egységára		0,203125 Ft	

Egy pillantást vetve a rendelésre, kiderül, hogy összeadások, szorzások és osztások végrehajtására van szükség. Lényeges dolog, hogyan rendezzük el a feladatot és a készülék milyen utasításokat kap.

Ha a programnak korlátlan számú cikket kell feldolgoznia, válasszuk a következő megoldási módot! Először döntünk el, hogyan írjuk be a változókat! Jelen példában a változókat a kijelzőn át írjuk be, miközben a program fut. Tehát nem az adattárolóban tároljuk, és később hívjuk be a változókat a programba. Ha a halmozódó végösszegeket számítjuk ki tároló aritmetikával, akkor minden adatsortot csak egyszer kell felhasználni, így azokat nem kell az adattárolóban tárolni. A közbenső adatok kijelzésével járó idővesztések csökkentésére a rendelt mennyiségeket halmozva tároljuk az R_1 -ben (az 1. adattárolóban R_1 , a 2.-ban R_2 stb.), a halmozott árösszeget az R_3 -ban, a mindenkor átlagos egységárat az R_4 -ben tároljuk. A mintaprogramot úgy készítettük el, hogy az egy árucikk összegét a mennyiség és az egységár beadása után azonnal megtudjuk, a többi eredmény tetszőlegesen behívható.

Befejezésül megjegyezzük, hogy a programnak egy bevezető programrészt kell tartalmaznia, mely az R_1 és R_2 adattárolót nullára állítja, mivel az R_1 és R_2 kiindulási művelete összeadás lesz. Ebben a példában érthetővé válik a megoldás megszervezésének fontossága. A vázolt megoldás segítségével tetszőleges számú tételt lehet feldolgozni.

Most készítsuk el a folyamatábrát, és határozzuk meg a billentyűzeti utasításokat!



PROGRAMUTASÍTÁSOK				
Lépés	Eljárás	Beírás	Billentyű	Kijelzés
1	a programtároló törlése és programmutató visszaállítása		2nd CP	0.
2	a tanulási üzemmód beadása		LRN	000 00
3	átszámítási program beadása		LRN	0.
4	a tanulási üzemmód feloldása		E	0.00
5	programkezdés		A	
6	az egyes árucikkek rendelt mennyiségének beadása	mennyiség	R/S	mennyiség
7	egységár beadása	egységár		a rendelt áruk összára
	A 6. és 7. lépést minden árucikknél megismételni!			
	Egy cikk beadása után a következő változókat jelezhetjük ki:			
	halmozott rendelés		B	rendelt mennyiség
	ár		C	ár
	átlagos egységár		D	átlagár

Tárolási hely és

billentyűkód Billentyűzés

000 76	2nd Lbl
001 15	E
002 47	2nd CMs
003 25	CLR
004 58	2nd Fix
005 02	2
006 91	R/S
007 76	2nd Lbl
008 11	A
009 44	SUM
010 01	1
011 42	STO
012 04	4
013 91	R/S
014 49	2nd Prd
015 04	4
016 43	RCL
017 04	4
018 44	SUM
019 02	2
020 43	RCL
021 02	2
022 55	÷
023 43	RCL

Tárolási hely és

billentyűkód Billentyűzés

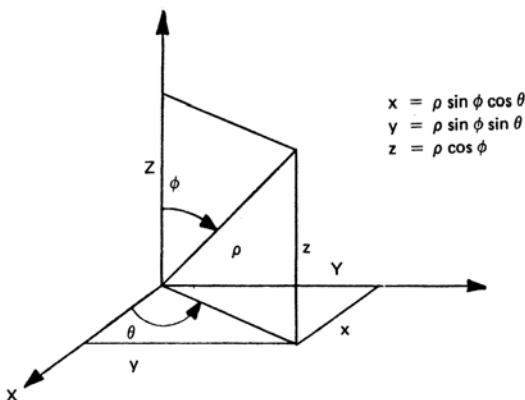
024 01	1
025 95	=
026 42	STO
027 03	3
028 43	RCL
029 04	4
030 91	R/S
031 76	2nd Lbl
032 12	B
033 43	RCL
034 01	1
035 91	R/S
036 76	2nd Lbl
037 13	C
038 43	RCL
039 02	2
040 91	R/S
041 76	2nd Lbl
042 14	D
043 43	RCL
044 03	3
045 91	R/S

A program végrehajtása a kezdetben megadott adatokkal:

Beírás	Billentyű	Kijelzés	Megjegyzések
100	E	0.00	bevezetés
.25	A	100.00	rendelt mennyiség (1)
	R/S	25.00	egységár (1)
200	A	200.00	a rendelt árucikk összára
15	R/S	30.00	a rendelt árucikk mennyisége (2)
.35	A	50.00	egységár (2)
50	R/S	17.50	a rendelt árucikk összára
.4	A	150.00	rendelt mennyiség (3)
	R/S	60.00	egységár (3)
300	A	300.00	a rendelt árucikk összára
1	R/S	30.00	rendelt mennyiség (4)
	B	800.00	egységár (4)
	C	162.50	a rendelt árucikk összára
	D	0.20	rendelt mennyiség (5)
	INV 2nd Fix	0.203125	egységár (5)
			a rendelt árucikk összára
			a rendelt mennyiség halmozott értéke
			halmozott ár
			átlagos egységár kerekítve
			átlagos egységár nem kerekítve

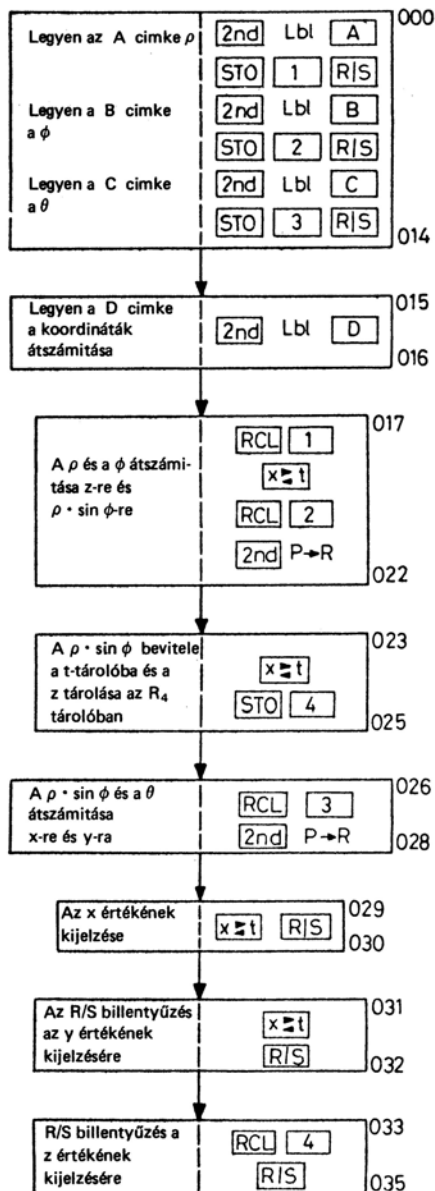
Program gömbkoordináták átszámítására

Írjunk programot gömbkoordináták derékszögű koordinátákra való átszámításához, az alábbi összefüggések alapján:



A készüléknek a 2nd P → R billentyűzéssel a fenti átszámításhoz kész programja van (lásd az V. fejezetet). Ez itt is hasznosítható.

A gömbkoordináták beírásának legegyszerűbb módja, a ρ , a ϕ és a θ értékek tárolása az R₁, az R₂, és az R₃ adattárolóba. Ha arra utasítjuk a gépet, hogy a ρ -t a t-tárolóba, és a ϕ -t a kijelző tárolójába vigye be, akkor z-t a 2nd P → R billentyűzéssel át lehet számítani derékszögű koordinátájú értékre. Ez az átszámítás a $\rho \cdot \sin \phi \cdot t$ a kijelzőtárolóba, és $\rho \cdot \cos \phi = z$ -t a t-tárolóba viszi. Ha a z-t az R₄-ben tároljuk és a $\rho \cdot \sin \phi \cdot t$ a t-tárolóba visszük, újra alkalmazhatjuk azt az átszámítást, hogy a θ behívása után a kijelző-tárolóba az x és az y értéket meghatározzuk. A programot úgy kell kialakítani, hogy x, y és z – ebben a sorrendben – az R/S billentyűvel kiszámítható legyen.



PROGRAMUTASÍTÁSOK				
Lépés	Tevékenység	Beírás	Billentyű	Kijelzés
1.	A programtároló törlése és programmutató visszaállítás		2nd CP	0.
2.	Átállás tanulási üzemmódra		LRN	000 00
3.	Programbeírás			
4.	A tanulási üzemmód feloldása		LRN	0.
5.	A ρ beírása	ρ	A	ρ
6.	A ϕ beírása	ϕ	B	ϕ
7.	A θ beírása	θ	C	θ
8A	A koordináták számítása és az x kijelzése		D	x
8B	Az y kijelzése		R/S	y
8C	A z kijelzése		R/S	z

Tárolási hely és

billentyűkód

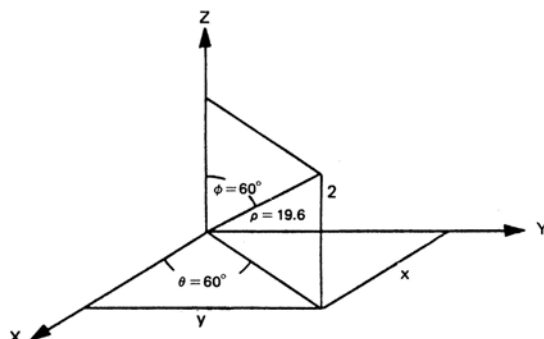
000 76	2nd Lbl
001 11	A
002 42	STO
003 01	1
004 91	R/S
005 76	2nd Lbl
006 12	B
007 42	STO
008 02	2
009 91	R/S
010 76	2nd Lbl
011 13	C
012 42	STO
013 03	3
014 91	R/S
015 76	2nd Lbl
016 14	D
017 43	RCL
018 01	1

Tárolási hely és

billentyűkód

019 32	$x \leftrightarrow t$
020 43	RCL
021 02	2
022 37	2nd P→R
023 32	$x \leftrightarrow t$
024 42	STO
025 04	4
026 43	RCL
027 03	3
028 37	2nd P→R
029 32	$x \leftrightarrow t$
030 91	R/S
031 32	$x \leftrightarrow t$
032 91	R/S
033 43	RCL
034 04	4
035 91	R/S

Példa: Számítsuk át a következő gömb-koordinátákat derékszögűekre: $\rho = 19.6$, $\phi = 60^\circ$, $\theta = 60^\circ$.



Beírás	Billentyű	Kijelzés	Megjegyzés
	2nd Deg		Szögegység = (régi) fok
19.6	A	19.6	ρ
60	B	60	ϕ
60	C	60	θ
	D	8.487048957	x
	R/S	14.7	y
	R/S	9.8	z

HALADÓ PROGRAMOZÁSI FELADATOK

További részletek a címkekről

Mint tudjuk, az A...E és az A'...E' billentyűket mint címkeket használhatjuk. Amennyiben egy programszakaszt ezen billentyűk valamelyikével megjelöltünk, a programmutató erre a programrészre áll rá, és a végrehajtás azonnal megkezdődik, mielőtt a szóbanforgó billentyűk valamelyikét megnyomjuk.

E programcím-billentyűkön túlmenően a készülék csaknem minden elsődleges és másodlagos rendeltetésű billentyűjét használhatjuk címkeként, pl. a 2nd sin, x^2 , =, CLR, 2nd Pgm, EE, 2nd Fix és a többiek is lehetnek címkek, hogy csak néhány példát említsünk. Ezeket a címkeket általános címkeknek nevezzük. A következő billentyűk kivételek, ezeket nem lehet címkeként alkalmazni:

Lbl, 2nd, SST, Ins, LRN, BST, Del, Ind, továbbá a 0...9 számjegyeket.

Az általános címkek és a programcímkek közt az a különbség, hogy az általános címke a programot nem tudja indítani, ha a címke által meghatározott utasítást kézzel billentyűzzük be. Ha egy programszakaszt x^2 -tel megjelöltünk, a kijelzett értéket a gép négyzetre emeli, ha x^2 jelű billentyűt megnyomjuk. A GTO x^2 R/S billentyűzése esetén a programfuttatás az x^2 címkekével kezdődik. Így most már több mint 60 címke áll rendelkezésünkre.

A programcím címkekhez hasonlóan, az általános címkeket bárhol beilleszthetjük a programba. A nyilvánvalóan összetartozó utasításcsoportokat mint pl. STO 12 vagy 2nd Fix 6 vagy Inv 2nd sin, nem szabad elválasztani! Jelen fejezet hátralevő részében a dolog sokféleségére való tekintettel ismertetjük a programcím billentyűknek címkeként való felhasználását.

Elágazási utasítás

Néhány további lényeges utasítás növeli számológépünk programozási kapacitását. Növelik a készülék rugalmasságát az utasítások feldolgozási sorrendjének vezérlésénél az elágazási utasítások vagy egyszerűn elágazások. Egy program főágát az átmenettel egy másik tárolóhelyre tudjuk átváltítani. Közvetlen és közvetett elágaztatást különböztetünk meg.

A közvetlen, ill. feltétlen elágazási utasítások (átmenetek) előfeltétel nélküli átmenetet hoznak létre ott, ahol erre szükség van. Ezek a számítástól függetlenek. Ezzel szemben a közvetett, ill. feltételhez kötött elágazási utasítás esetén a program, valamilyen követelménytől függően hozza létre az átmenetet.

FELTÉTEL NÉLKÜLI ELÁGAZÁSOK

Feltétel nélküli elágazási utasítások a következők: RST, GTO és SBR. Az RST programmutatót automatikusan a 000 tárolóhelyre állítja be. A GTO és az SBR utasítás a programmutatót minden további feltétel nélkül a megadott tárolóhelyre irányítja. Ügyeljünk arra, hogy az RST-nek még más rendeltetései is vannak (lásd a programvezérlés alapfunkcióiról szóló részt az V. fejezetben).

A „go-to” utasítás GTO

A programok átdolgozása c. fejezetben már tárgyaltuk a GTO utasítást; abban az esetben, ha kézi billentyűzéssel írjuk be. Hasonlóan működik az utasítás a program végrehajtása alatt. A GTO utasítás a programmutatót közvetlenül a megadott tárolóhelyre irányítja, abban az esetben, ha abszolút cím vagy címke követi. A program az új tárolóhelynél folytatódik.

Rövidített alakú címzés is alkalmazható abszolút címeknél. Tanulási üzemmódban a GTO 9 akkor, és csak akkor azonos a GTO 009-cel, ha a következő billentyűutasítás nem számjegy. Írjuk be a következő rövid számolási programot!

Billentyű	Kijelzés	Megjegyzés
CLR 2nd CP	0.	egy program előkészítése
GTO 9 LRN	009 00	átmenet a 009 tárolási helyre és a tanulási üzemmódba való átváltás
+	010 00	programbeadás
1	011 00	
=	012 00	
2nd Pause	013 00	
GTO	014 00	
9	014 00	a tárolóhely száma nem ugrik, mert a cím hátralevő részét várja
LRN	0.	a tanulási üzemmód feloldása
LRN	016 00	a programmutató továbblépett,
LRN	0.	mivel az LRN utasítással befejeződött a címzés.

E program végrehajtásához nyomjuk be a GTO 9 R/S billentyűket!

A programtárolóban az abszolút címek 2 tárolóhelyet foglalnak el. A GTO 136 utasításhoz tehát 3 tárolási helyre van szükség. Az első hely a GTO utasítást, a másik kettő a címet tartalmazza. Az utóbbiak közül az első helyre a cím százasai kerülnek, a tízesek, egyesek pedig összefoglalva a következő tárolási helyre. Az eredő billentyűzés 61 01 36 a programtárolóban. Ennél a tömör tárolási módnál kombinált utasításokról beszélünk, ahol is a kódkombinációkat teljesen automatikusan a számológép hozza létre. Végezzük el a következő gyakorlatot:

Billentyű	Kijelzés	Megjegyzés
2nd CP CLR	0.	a programtár és a kijelző törlése
GTO 136	0.	a programmutató beállítás a 136. tárolóhelyre
LRN	136 00	
R/S	137 00	R/S elfoglalja a 136 tárolóhelyet
LRN RST LRN	000 00	visszatérés 000-ra
GTO	001 00	GTO elfoglalja a 000 tárolóhelyet
1	001 00	
3	001 00	
6	003 00	a programtár megvárja mindhárom értéket, mielőtt a címet befogadná
BST	002 36	
BST	001 01	a cím a korábbiaknak megfelelően tárolódik
BST	000 61	
LRN	0.	
R/S	0.	a program végrehajtása
LRN	137 00	azonnali átmenet a 136. tárolóhelyre, ahol az itt tárolt R/S utasítást hajtja végre. A programmutató beáll a 137. tárolóhelyre.

Egy címkének ugyanez lenne a hatása. Ha pl. a GTO x^2 utasítást kézi billentyűzéssel írjuk be, az x^2 címkére áll be a programmutató és a készülék várja a további utasítást. Ha utasítjuk a készüléket, hogy egy nem létező címkét keressen meg, a pillanatnyi érték villogni fog a kijelzőben.

Ezekből a példákban is kitűnik GTO utasítás szerepe, ha azt billentyűzve adjuk be, és ha az része egy programnak. Más elágazásoknak is teljesen azonos a hatása. Ha kézi billentyűzéssel írjuk be, vagy ha valamely programban fordulnak elő, az átmenet a megfelelő tárolóhelyre történik.

Alprogramok

Gyakran találkozunk olyan műveletekkel, melyeket ismételten többször egymás után kell végrehajtani. Ezeket a műveleteket alprogramoknak nevezzük. Az alprogramok lehetőséget adnak arra, hogy olyan, ún. mellékfolyamatot vagy billentyűutasítást hajtunk végre, mely valamilyen különleges célt szolgál. Ezeket megjelölhetjük, és bármikor, bármely helyről visszanyúlhatunk hozzájuk. Ez a folyamat majdnem olyan egyszerű, mintha egy billentyűt nyomnánk meg. Az alprogram lezárása után a programmutató automatikusan arra a tárolóhelyre irányul vissza, mely az elágazási pontot követi. Ha alprogramot használunk, gyakran ennek a programrésznek behívásáról beszélünk. Ezzel utasítjuk a készüléket arra, hogy egyetlen alprogram-utasítással lépések egész sorát hajtja végre.

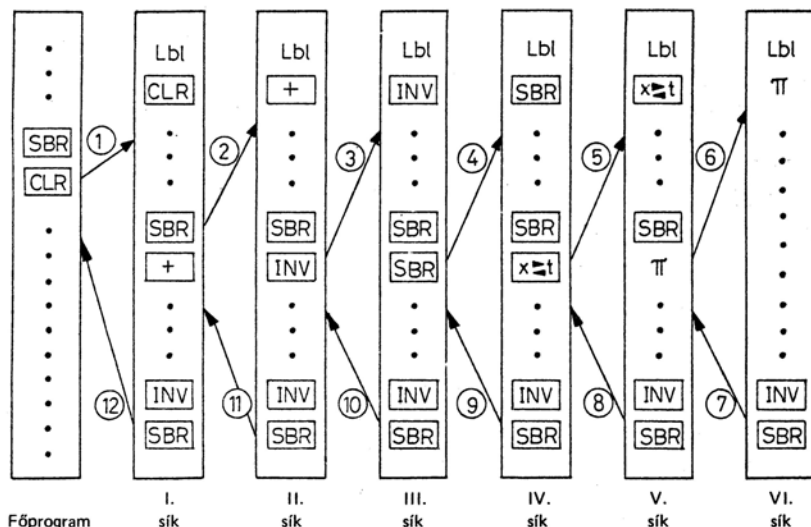
A célszerű programozási technika érdekében a programokat alprogramként alkalmazhatóan szerkesztjük. Így azokat változtatás nélkül, más programoknál is felhasználhatjuk. Ennek érdekében, ha a program futását meg akarjuk szakítani, az R/S utasítás helyett írjunk be INV SBR utasítást. Jelen fejezet programjait ilyen formában szerkesztettük.

Az alprogramutasítás SBR

Az alprogramutasítás kétszeresen módosított átmeneti utasítás. A billentyűzet segítségével elindítható a program végrehajtása és a program közben ellenőrzi, hogy melyik helyről induljon az elágazás. Ha az SBR 136 utasítást kézi billentyűzéssel írjuk be, a programmutató azonnal beáll a 136. tárolóhelyre és a végrehajtás automatikusan ezen a helyen kezdődik. A hatás azonos a következők billentyűzésével: GTO 136 R/S. Ugyanez a folyamat jön létre az SBR x^2 megnyomásával. A program végrehajtása az x^2 címkénél kezdődik, függetlenül attól, hogy a cím hol található.

Ha az SBR 136 vagy a GTO 136 utasítást az előző példában a 000...002 tárolóhelyekre adtuk volna be, a művelet sor ugyanúgy ment volna végbe. Ha az alprogram lefut, a 003 programcím az alprogramvisszaugrató tárolóban tárolódik. Ha tehát egy számtáblasorozat, amely INV SBR utasítással végződik, 136-nál kezdődik, a feldolgozás során vezérli az alprogramvisszaugrató tárolót, ahol 003 tárolódik. A folyamat tehát itt folytatódik. Zárjuk le az alprogramot INV SBR utasítással hogy a végrehajtást újra a kiindulási pontra irányítsuk! A tényleges folyamat 136-nál elágazással tért át egy alprogramra, majd automatikusan visszatért. Ez utóbbit a programban az INV SBR utasítással értük el.

Az alprogramvisszaugrató tároló legfeljebb 6 visszaugratási címet képes egyidejűleg tárolni. Ez azt jelenti, hogy egy alprogram maga is tartalmazhat alprogramot, azt be is hívhatja, ez utóbbi szintén használhat alprogramot és így tovább egészen a hatodik alkalomig.



Ha az ábra szerinti lépéseket egy program részeként írtuk volna be, a feldolgozás az ábra szerinti számozás sorrendjében történt volna. Figyeljük meg, hogy minden alprogram INV SBR utasítással zárul! Ezzel a gépet arra utasítjuk, hogy az alprogramvisszaugrató tárolóból az utolsó tárolt címet vegye elő és oda térjen vissza. Általában a feldolgozási folyamat a főprogramban végződik, amelyben az első alprogram behívással kezdődött. Egyébként az INV SBR utasítás a programtárolóban kombinálódik és 92 billentyűkóddal, és csak egy tárolóhelyet foglal el. Ez a kód nem vezethető le a sor-oszlop hozzárendeléséből.

Ha egy programrészt programcím-billentyűvel jelöltünk, akkor ez a rész a billentyűzet segítségével hajtható végre, és csak a megfelelő címkét kell megnyomni. A folyamat akkor jön létre, ha ezen billentyűk egyike a programnak része; a programmutató beáll a címkére és a végrehajtás folytatódik. Ezek a programcím-billentyűk automatikus belső SBR utasítást tartalmaznak. Ha tehát egy programszakaszt programcím-billentyűvel jelöltünk meg, és INV SBR utasítással zárunk, akkor ezt a részt a készülék úgy dolgozza fel, mintha SBR utasítással hívtuk volna be.

Az alprogramok behívása

Meghatározásszerűen az alprogram olyan programszakasz, amelyet meghatározott célból hoztunk létre; csak egyszer kell beírni, de ismételten alkalmazható. Az alprogramokat INV SBR utasítással kell befejezni, hogy a feldolgozási folyamatot újra a behívandó utasításcsoportra irányítsuk. Alprogramok behívására három mód-szer létezik:

1. abszolút cím (SBR 136)
2. általános címke (SBR x^2)
3. programcímke (A)

Az alprogramok címkével való megjelölése a programutasításokat áttekinthetővé teszi. Sőt, alkalmazhatunk olyan címkéket is, amelyek az alprogram célját megjelölik. A címkéket körültekintéssel válasszuk meg, és jegyezzük fel azok feladatát! A megjelölt alprogramokat tetszőleges helyen vihetjük be a programtárolóba, mert behíváskor a címkéi a gép a programtárolóban elfoglalt helyétől függetlenül megtalálja. Lényeges továbbá, hogy a címkézett alprogramot az Ins és a Del utasítás nem befolyásolja. Ezeket a készülék a prog-

ramtárolóban az alprogram előtt hajtja végre.

Olyan programhoz, amellyel az $x^2 + 3x$ kifejezést beadandó x -értékek mellett számítja ki, a következő billentyűket kell működtetni:

STO
1
 x^2
+
3
X
RCL
1
=
R/S

A program a következő

A program főresze:

000 2nd Lbl
001 A
002 SBR
003 STO
004 x^2
005 +
006 3
007 X
008 SBR
009 RCL
010 =
011 R/S

Alprogramok:

012	2nd Lbl
013	STO
014	STO
015	01
016	INV SBR
017	2nd Lbl
019	RCL
019	RCL
020	01
021	INV SBR

Írjunk be egy tetszőleges x értéket és nyomjuk meg az A billentyűt! A feldolgozási folyamatot nyilakkal ábráztuk. Figyeljük meg, hogy a programindítás a legegyszerűbb módon történt, vagyis programcímke alkalmaztunk. Az alprogramcímet úgy választottuk meg, hogy az egyes programszakaszok azonosíthatók, de más címkeket is alkalmazhattunk volna.

Az alprogramok használatánál a következőkre kell tekintettel lenni.

Az RST és az = az a két utasítás, amelyeket alprogramoknál óvatosan kell alkalmazni. Emellett biztosítanunk kell azt, hogy új feladat megkezdése előtt a készülék törölje az alprogram-visszaugrató tárolót.

A reset utasítás egyéb rendeltetései mellett automatikusan törli az alprogram-visszaugrató tárolót. Ha egy alprogramban vissza kell mennünk a 000 tárolási helyre (az RST billentyű elsődleges rendeltetése), használjunk GTO 000 utasítást vagy egy címke, ha a 000-ban címke tároltunk! Az = egyenlőségutasítás az összes befejezetlen műveletet elvégzi. Ha ezt az utasítást az alprogramban alkalmazzuk, nemcsak az alprogram befejezetlen műveleteit végzi el, hanem a főprogramét is.

Vizsgáljuk meg a következő programrészletet, amely a $4 + (1+2) \cdot 3$ kiszámítására szolgál:

4
+
SBR x^2
X
3
=
R/S
2nd Lbl x^2
1
+
2
=
INV SBR

Itt az x^2 jelű alprogramban az egyenlőségutasítás a $\times 3$ -hoz való visszatérés előtt nemcsak az $1 + 2$ műveletet, hanem a $4 +$ műveletet is elvégzi. Emiatt eredményül 21-t kapunk, bár a helyes eredmény 13

A program könnyen megváltoztatható úgy, hogy helyes megoldáshoz jutunk. Használjunk zárójelet az alprogram számításához!

```
4
+
SBR x2
X
3
=
R/S
2nd Lbl x2
(
1
+
2
)
```

INV SBR. Ez a folyamat a helyes eredményt, a 13-t adja.

Ajánlatos megjegyezni, hogy minden alprogramot bal zárójellel kezdünk és jobb zárójellel zárunk az INV SBR utasítás előtt! Zárójeleknél eggyel több billentyűzésre van szükség, mint az $=$ esetén, de így sok hibás számítást kerüldünk el. A voltaképpeni előny az, hogy zárójeleknél csak a zárójelpáron belüli befejezetlen műveleteket végezzük el.

Ha ilyen esetekben az egyenlőségutasítást mellőzzük, a feladat semmiképp sem válik nehezebbé. A zárójelek arra szolgálnak, hogy ilyen kifejezéseket másoktól különválasztva számíthassunk. Néhány értékes információt azonban figyelembe kell venni, ha a kijelzőtároló pillanatnyi értékét fel akarjuk használni.

Ha egy alprogramnak a behívás pillanatában a kijelzőtároló tartalmához ismételten hozzá kell nyúlnia, akkor a számítások végrehajtása előtt a változót egy adattárolóban kell tárolni és a felhasználáskor behívni. Ha a kijelzőtároló tartalmára csak a számítás elején van szükség, a CE billentyűvel olyan megoldást alkalmazhatunk amivel ezt az értéket a zárójeles kifejezésbe vihetjük be. Ez egy program részeként ugyanúgy működik, mint kézi billentyűzésnél a billentyűzet.

Billentyűzzük a következőket! $2.18 \times (CE + 6) =$. Kijelzés: 17.8324.

A fenti utasítássorozatban a CE billentyű a 2,18 értékét beviszi a zárójeles kifejezésbe, és lehetővé teszi a készüléknek a $2,18 \cdot (2,18 + 6) = 17.9324$ számítás elvégzését.

Alkalmadtán valószínűleg olyan programot is fogunk készíteni, amely egy alprogramon belül befejeződik. Más szavakkal, a feladat eredményét úgy kapjuk meg, hogy a vezérlést a behívandó programnak nem adjuk vissza. Ilyen esetekben a vezérlés visszaadása nyitva marad, mert nem töröljük az alprogram-visszaugrató tárolót. Ha a számológépet nem kapcsoljuk ki, vagy ha az RST billentyűt nem nyomjuk meg, vagy ha a visszaugrató-tárolót a 2nd CP utasítással nem töröljük, akkor az új feladat végrehajtásakor nehézségek léphetnek fel, mivel a program a korábbi visszaugratási címekhez hibásan térhet vissza. Annak érdekében, hogy egyes feladatok hibás megoldását a nyitvahagyott visszaugratási címekből adódóan elkerüljük, az alprogram-visszaugrató tárolót az RST utasítással alapvetően törölni kell. A törlést kézzel is végezhetjük; előnyösebb azonban, ha az RST utasítást a megfelelő helyen a programba iktatjuk.

Software-programok, mint alprogramok

Előfordulhat, hogy a saját készítésű programot célszerű egy software-program beiktatásával bővíteni. Ugyanaz a billentyűzés, amely a software-programokat hozzáférhetővé teszi, módot ad azoknak a saját programba való beiktatására. Így a software-programok egyszerűen a saját programjaink alprogramjai lesznek a tárolóban. A software-egységek mindegyikében olyan utasítások ezreit tároljuk, amelyek alprogramként is használhatók. Minden software-program maga is alprogram. Ezekben – kevés kivétellel – nincs $=$ vagy RST utasítás; az alprogramokat zárójelekkel összegezzük és INV SBR utasítással zárjuk le.

Ha megnyomjuk a 2nd Pgm mm billentyűket, a programmutató beáll a software-program mm számára.

Programban a Pgm művelet hasonló az SBR utasításhoz. Az egyetlen különbség az, hogy a 2nd Pgm mm utasítás arra utasítja a gépet, hogy az alprogramot az mm software-programban keresse, és ne magában a gép programtárolójában. Amint a software-program befejeződik, a programmutató visszatér a programtárolóban levő behívási helyre és a folyamat a szokásos módon folytatódik. A kétjegyű programszámot a gép összekapcsolja, így az egyetlen tárolási helyet foglal el.

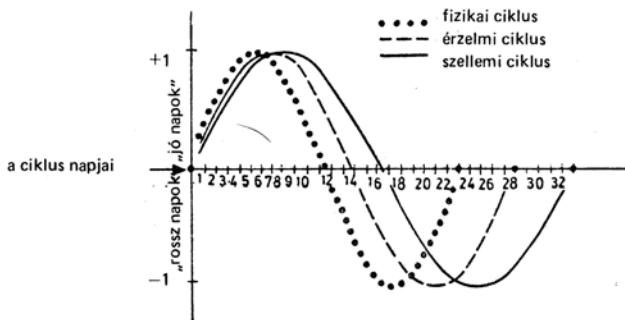
Ha a software-program egy szakaszát programcím-billentyűvel jelöljük (pl. A-val), akkor a helyes billentyű-sorrend 2nd Pgm mm A lesz. Ha általános címkét alkalmazunk (pl. tan) a billentyűzés 2nd Pgm mm SBR 2nd tan. Vigyázat! Ha a 2nd Pgm mm után sem SBR utasítás, sem programcím-billentyű nem következik, akkor az eljárás nem használható, ill. hibás eredményekhez vezet!

Bioritmus-program

A bioritmus-elmélet hívei szerint 3 ciklus van az életünkben, amely születésünk napjával kezdődik:

1. a fizikai ciklus, 23 napos periódusokkal
2. az érzelmi ciklus 28 napos periódusokkal
3. a szellemi ciklus 33 napos periódusokkal.

Az egyes ciklusok első fele – állítólag – ún. jó napjainkat, második fele ún. rossz napjainkat tartalmazza.



A következő egyenlettel a bioritmus pillanatnyi hatását +1 és -1 közti értékként fejezhetjük ki:

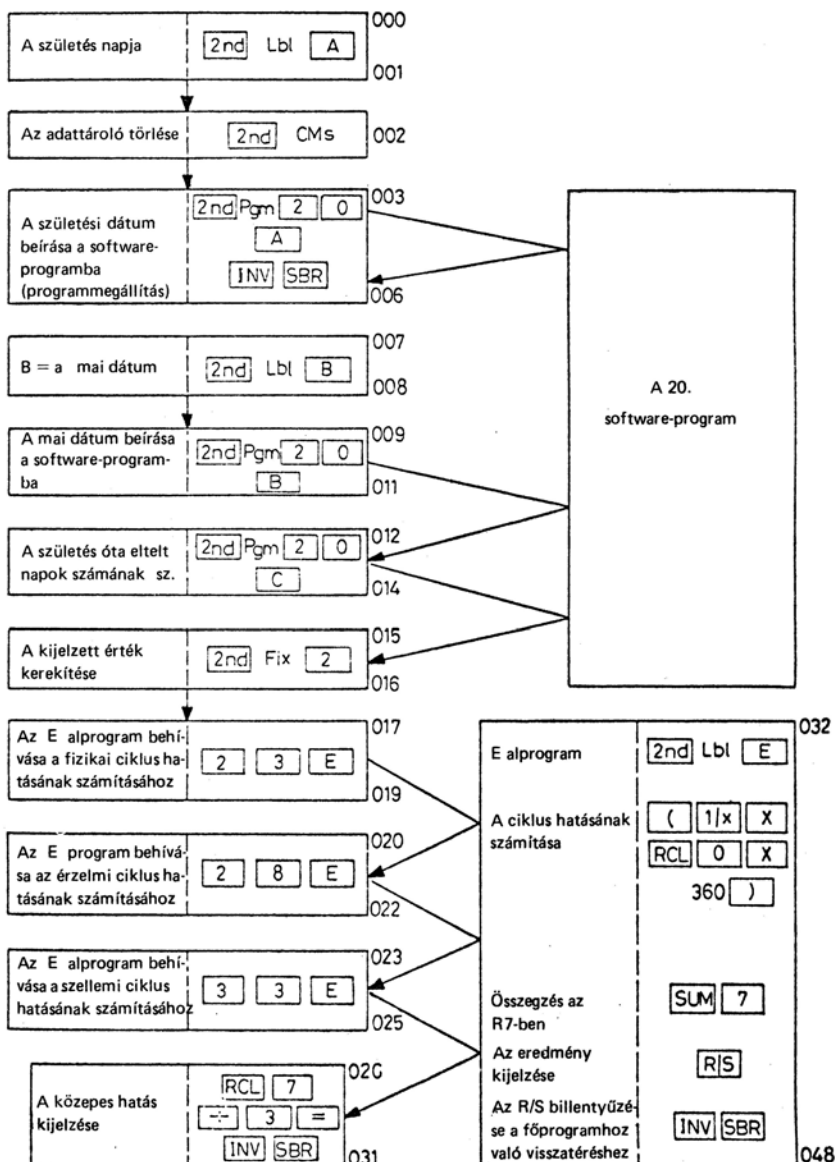
$$\text{amplitúdó} = \sin \left(\frac{\text{a napok száma születésünk óta}}{\text{a ciklus napjainak száma}} \cdot 360^\circ \right)$$

Ezek után írjunk olyan programot, amellyel adott személy bioritmusának ciklusai értékelhetők. Minthogy valamennyi ciklust ugyanaz az egyenlet írja le, a program egyszerűsítésére alprogramot alkalmazunk. Ezenfelül alprogramként elővesszük a 20. számút, hogy a születés óta eltelt napok számát kiszámítsuk.

A következő alprogramban az egyes ciklusok hatásának számítására az alprogramot az E címkével jelöljük, mivel a számítások közt az egyetlen különbség a ciklusnapok száma. A programot az R/S utasítással megszakíthatjuk, és a fizikai, az érzelmi és a szellemi ciklus hatását ebben a köztölt sorrendben jelezhetjük ki.

Az INV SBR utasítást itt nem használjuk, mert az a programvezérlést visszaadná a főprogramnak; ahelyett hogy megszakítaná a végrehajtást. (Lásd az Alprogram utasítás (SBR) c. részt néhány oldallal előbb!)

A program végeredménye a három érték középértéke. A készülék az összes kijelzett eredményt két tizedes értékre kerekíti.



Tárolóhely és billentyűkód	Billentyűzés	Tárolóhely és billentyűkód	Billentyűzés
000 76	2nd Lbl	027 15	E
001 11	A	028 43	RCL
002 47	2nd CMs	029 07	7
003 36	2nd Pgm	030 55	÷
004 20	2 0	031 03	3
005 11	A	032 95	=
006 92	INV SBR	033 92	INV SBR
007 76	2nd Lbl	034 76	2nd Lbl
008 12	B	035 15	E
009 36	2nd Pgm	036 53	(
010 20	2 0	037 35	1/x
011 12	B	038 65	X
012 36	2nd Pgm	039 43	RCL
013 20	2 0	040 00	0
014 13	C	041 65	X
015 42	STO	042 03	3
016 00	0	043 06	6
017 58	2nd Fix	044 00	0
018 02	2	045 54	)
019 02	2	046 38	2nd sin
020 03	3	047 44	SUM
021 15	E	048 07	7
022 02	2	049 91	R/S
023 08	8	050 92	INV SBR
024 15	E		
025 03	3		
026 03	3		

PROGRAMUTASÍTÁSOK				
Lépés	Eljárás	Beírás	Billentyű	Kijelzés
1.	a programtároló törlése és a programmutató visszaállítása	HHNNÉÉÉÉ HHNNÉÉÉÉ	2nd CP	000 00
2.	a tanulási üzemmód beírása		LRN	
3.	a bioritmus program beadás			
4.	a tanulási üzemmód feloldása		LRN	
5.	a születési dátum beírása		A	0. a fizikai ciklus hatása
6.	a jelenlegi dátum beírása és a fizikai ciklus hatásának számítása		B	
7.	érzelmi ciklus hatásának számítása		R/S	
8.	a szellemi ciklus hatásának számítása		R/S	a szellemi ciklus hatása
9.	a közepes hatás számítása		R/S	közepes hatás

Példa: XY 1944. május 2-án született. Számítsuk ki bioritmusát 1977. március 1-re. Feltételezzük, hogy a bioritmus-program meg a tárolóban van. HHNN.ÉÉÉÉ = hónap, nap, év.

Billentyű	Kijelzés	Megjegyzés
502.1944 A	0.	születési dátum beírása
301.1977 B	0.82	a fizikai ciklus hatása
R/S	1.00	az érzelmi ciklus hatása
R/S	0.76	a szellemi ciklus hatása
R/S	0.86	közepes hatás

Határozza meg ugyanezzel a programmal saját bioritmikus ciklusainak értékét! Ha saját programjainkban alprogramként software-programot alkalmazunk, különösen ügyeljünk arra, hogy melyik adattárolót foglalkoztatjuk le, és hogy a software-program melyik tárolót használja! Ha a két programban ugyanazt a tárolót különböző célokra használjuk fel, téves eredményhez jutunk.

FELTÉTELHEZ KÖTÖTT ELÁGAZÁSOK (DÖNTÉSI UTASÍTÁSOK)

A készülék egyéb előnyös tulajdonságai közé tartozik az utasítások azon csoportja, amelyek döntéshozatalra szolgálnak. Az ún. feltételhez kötött elágaztatási utasítások csoportja (feltételes átmeneti utasítások) teszlik lehetővé a döntéshozatalt. Valahányszor előfordul ezen utasítások egyike a programban, a gép először vizsgálatot végez, és ennek kimenetelétől függően dönt, hogy az átmenet megtörténik-e vagy sem.

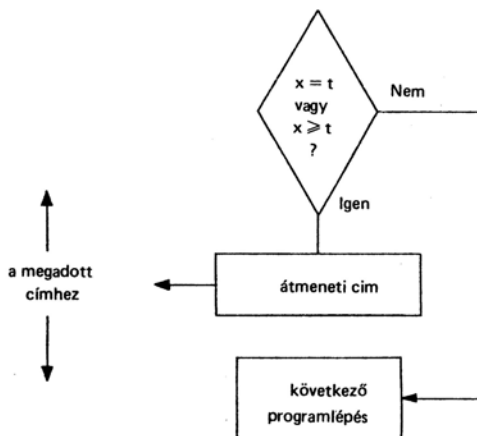
A feltételes átmeneti utasításoknak három fajtája van aszerint, hogy mi a vizsgálat tárgya.

1. A kijelzőtároló és a t-tároló tartalmának összehasonlítása: $x=t$, $x \geq t$, $x \neq t$, $x < t$.
2. A 0...9 adattárolók tartalmának vizsgálata: Dsz
3. A programjelző helyzetének vizsgálata: If flg.

Minden ilyen utasítás után elágazási címnek kell állnia. Erre a címre történik az átmenet, ha a vizsgálatra adott válasz igen (pozitív). Ha a vizsgálatra adott válasz negatív, akkor a készülék az elágazáson átugrik. Ha pl. a 2nd $x=t$ A vizsgálat eredménye pozitív, a program A-hoz lép, mintha GTO A utasítást adtunk volna be. Ha x nem egyenlő a t-tároló tartalmával, nem jön létre az átmenet.

A kijelző-tároló és a t-tároló összehasonlítása

Voltaképpen mi is a t-tároló? A t betű a test szó rövidítése, a tároló pedig olyan adatregiszter, ahol számokat tárolhatunk, behívhatunk és azokat a kijelzett számmal összehasonlíthatjuk. Az $x \neq t$ cserebillentyűvel vihetünk be számokat a t-tárolóba, ill. vihetünk ki onnan. Ez a billentyű egyszerűen felcseréli a kijelző-tárolóban levő x értéket A t-tárolóban levő t értékkel. Induláskor a t-tárolóban 0 van. Billentyűzzünk 5-öt és nyomjuk meg az $x \neq t$ billentyűt! Ekkor 0 lesz a kijelzőben, és 5 a t-tárolóban. Nyomjuk meg újra az $x \neq t$ billentyűt; akkor ismét 5 jelenik meg a kijelzőben, s közben a 0 visszakerül a t-tárolóba. A kijelző-tároló mindenkorinak és a t-tároló tartalmának összehasonlítására több utasítás alkalmas; az x egyenlő t-vel ($x=t$) és az x nagyobb vagy egyenlő t-vel ($x \geq t$). Minden ilyen utasítást abszolút programtárolócímmel vagy címkével kell követni. Ha egy vizsgálatot elvégzünk, (pl. $x=t$), és a válasz igen, akkor a program a feltételvizsgáló utasítás után álló címen folytatódik. Negatív válasz esetén a készülék a címet nem veszi tudomásul és a program úgy folytatódik, mintha nem is történt volna meg a vizsgálat. A folyamatot a következő ábra szemlélteti.

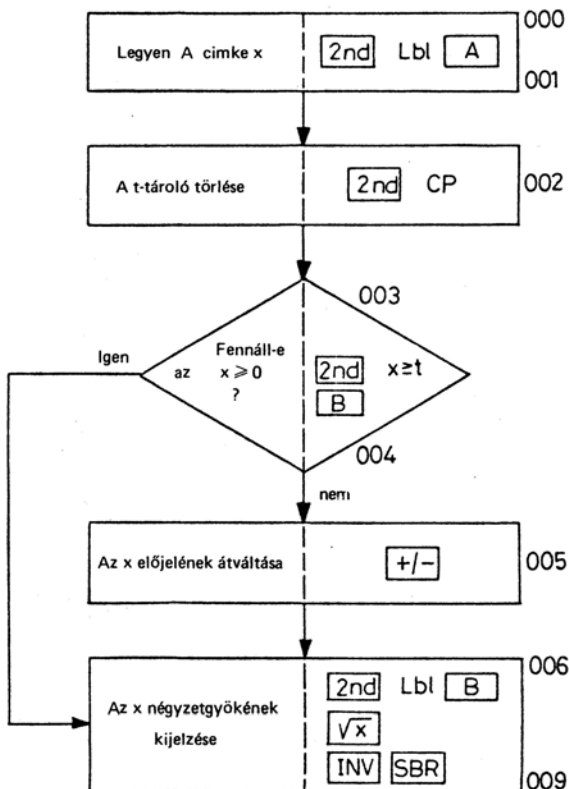


Ezek az utasítások elsősorban a programon belüli folyamat vezérlésére szolgálnak, de azokat a billentyűzettel is ugyanúgy végre lehet hajtani. Vizsgáljuk meg a következő billentyűutasításokat!

Billentyű	Kijelzés	Megjegyzés
5 $x \approx t$	0.	az 5-t a t-tárolóban tároljuk.
6 2nd $x \geq t$ 123	6.	a 6 beírása a kijelzőbe és utasítás a készülőleknek $x \geq t$ esetben átmenetre a 123. programtároló helyre
LRN	123 00	annak ellenőrzése, hogy az elágazás megtörtént-e, mivel $6 > 5$
LRN 4 2nd $x \geq t$ 111	4.	visszatérés a billentyűvezérléshez és vizsgálat $x = 4$ -esetre
LRN	123 000	még mindig a 123 tárolóhelyen vagyunk, mert $4 < 5$ miatt nem jött létre átmenet
LRN 5 2nd $x \geq t$ 111	5.	visszatérés a billentyűzéshez és vizsgálat $4 = 5$ esetre
LRN	111 00	most jön létre az átmenet, mert a kijelzés egyenlő a t-tároló 5 értékével.

Négyzetgyökvonás

Az a feladat, hogy számítsuk ki, a kijelzőbe beadott tetszőleges x szám abszolút értékének négyzetgyökét. Ha a szám negatív, váltsuk át az előjelét, és ezután vonjunk gyököt!



Tárolóhely és
billentyűkód

000 76
001 11
002 29
003 77
004 12
005 94
006 76
007 12
008 34
009 92

Billentyűzés

2nd Lbl
A
2nd CP
2nd $x \geq t$
B
+/-
2nd Lbl
B
 $\sqrt{x}$
INV SBR

Most már végrehajthatjuk a programot. Írjuk be pl. a 4 értéket, és nyomjuk le az A billentyűt! Az eredmény 2. Ha -4-t adunk be, és megnyomjuk A-t, ugyanezt az eredményt kapjuk.

Ezeket a vizsgálati utasításokat az INV billentyűvel is alkalmazhatjuk, az elágazási feltételek megfordítására. Lássuk a következő példát:

Utasítások

2nd $x=t$
 INV 2nd $x=t$
 2nd $x>t$

INV 2nd $x>t$

Kérdésfeltevés (teszt)

A kijelzőtárolóban levő érték egyenlő a t-tárolóban lévővel?
 A kijelzőtárolóban levő érték nem egyenlő a t-tárolóban lévővel?
 A kijelzőtárolóban levő érték egyenlő a t-tárolóban lévővel vagy annál nagyobb?
 A kijelzőtárolóban levő érték kisebb mint a t-tárolóban lévő érték?

Amennyiben a fenti kérdések bármelyikére adott válasz igen, a program azon a címen folytatódik, amelyet közvetlenül ezen utasítás után adunk meg. Ha a válasz nem, akkor a beiktatott címet a készülék nem veszi tudomásul és a program a programtároló következő tárolóhelyénél folytatódik.

A flag (jelző) művelet

Mi a jelző és mi a szerepe a programban? A jelző egy belső kapcsolórendszer, ki- vagy bekapcsolt állapotban. Képletesen kifejezve, a programjelző vagy szabad jelzést ad, vagy tiltót. A jelző valamely ponton beilleszthető a programba, és egy későbbi időpontban vizsgálható. A jelzőnek ez a léptetése és visszaléptetése, azaz a szabad út jelzése vagy tiltása ill. a jelzők vizsgálata a kijelzőtárolótól és az adattárolótól függetlenül megy végbe.

A jelzők sokféle alkalmazása közül most bemutatunk néhányat.

1. Program, előválasztások vezérlése a program végrehajtása előtt, billentyűzéssel
2. A programfeltételek jelzõt képeznek a későbbi vizsgálatokhoz.
3. A program végrehajtásának ellenőrzése, vagyis hogyan vezetett a program a pillanatnyi helyzethez.

10 külön jelző áll rendelkezésünkre, számosságuk 0...9. Következésképp egy-egy jelzőutastításhoz meg kell adnunk, hogy melyik jelzőről van szó. A jelzők vezérlésére az alábbi utasítások szolgálnak.

1. 2nd St flg y — az y-jelző beléptetése
2. INV 2nd St flg y — az y-jelző visszaléptetése, egy meghatározott kimeneti állapot létrehozása
3. 2nd If flg y — az y-jelző vizsgálata és elágazás, ha a jelző szabad utat jelez. Ezt az utastítást a korábban tárgyalt vizsgálati utasításokhoz hasonlóan elágazási címmel kell kiegészíteni.
4. INV 2nd St flg y utastítás az utána következő elágazási címmel, az y-jelző vizsgálata és elágaztatás, ha az tiltó állásban van.

Ezeket az utastításokat vagy billentyűzéssel adjuk be, vagy azok részei lehetnek egy programnak. Billentyűzük a következőket, és figyeljük meg a jelzők hatását!

Billentyű	Kijelzés
2nd CP	0.
2nd St flg 4	0.
2nd If flg 4 136	0.
LRN	136 00
LRN 2nd If flg 5 111	0.
LRN	136 00
LRN INV 2nd St flg 4	0.
2nd If flg 4 222	0.
LRN	136 00
LRN INV 2nd If flg 4 222	0.
LRN	222 00

Megjegyzések

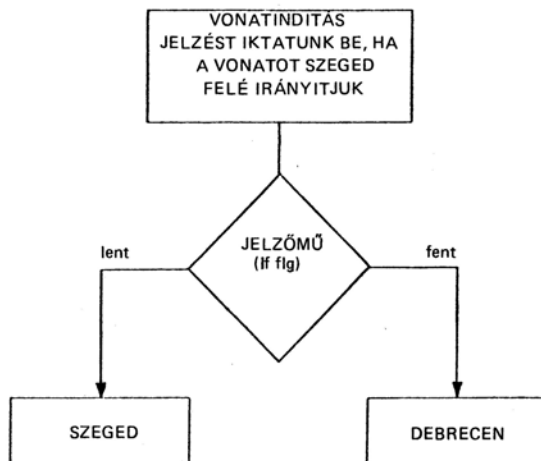
a kijelző és a programtároló törlése: ezzel az utastítással valamennyi programjelzõt és a t-tárolót is töröljük
 a 4-es jelzõt beléptetése
 a 4-es jelzõ vizsgálata; ha léptettük, átmenet a 136.tárolóhelyre.
 létrejött az átmenet a 136-hoz, mert a jelzõt beléptettük
 az 5-ös jelzõ vizsgálata; ha léptettük, átmenet a 111.tárolóhelyre.
 átmenet nem jött létre, mert a jelzõt nem létettük be
 a 4. jelzõt visszaléptettük
 a 4. jelzõ vizsgálata; ha léptetve van, átmenet a 222.tárolóhelyre
 a jelzõ nincs léptetve, tehát nem jön létre átmenet
 a 4. jelzõ vizsgálata; ha nincs léptetve,átmenet a 222. helyre
 a jelzõ nincs léptetve, tehát program átmenet a 222. felé

A jelző-vizsgáló utasítás hasonló a t-tároló vizsgáló utasításhoz. A különbség az, hogy előbbiek a jelzők állapotát vizsgálják, míg utóbbiaknál a kijelző és a t-tároló tartalmát hasonlítjuk össze. Az utasítás utáni elágazási cím abszolút cím lehet, mint a fenti példában is; de lehet tetszőleges típusú címke is, programcímke vagy általános címke.

Egy, már beiktatott jelző léptetése, vagy egy már kiiktatott jelző visszaléptetése nincs hatással sem a jelző állapotára, sem a számításokra. Minden jelzőt azonnal visszaléptethetünk az RST vagy 2nd CP utasítással. Ügyeljünk ezenkívül arra, hogy a programjelzők nem figyelhetők meg a kijelzőben közvetlenül mint a t- vagy más tároló esetében, azaz hogy a jelzőt léptettük-e vagy visszaléptettük. A jelző állapotát csak a vizsgálat érzékelteti. A következő helyzet szemlélteti a jelzők alkalmazásának célját és módját.

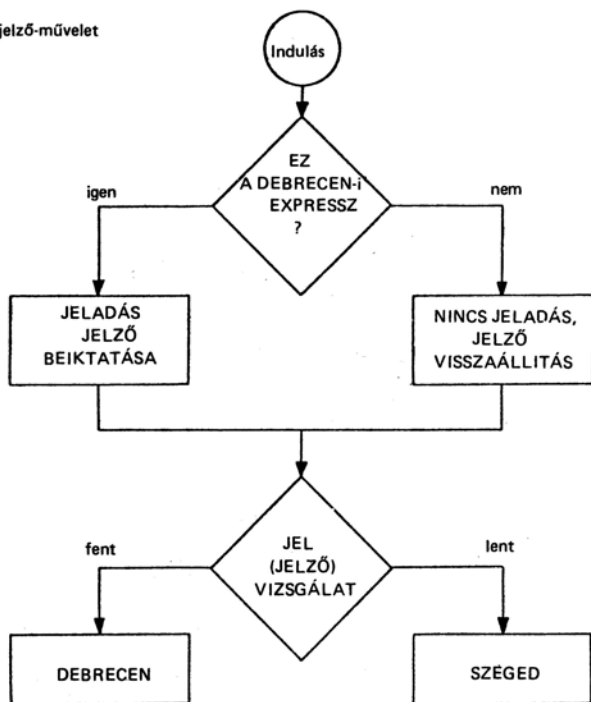
A programjelző szerepe hasonló a vasúti jelzőműhöz, melynek alsó vagy felső helyzetétől függ a vonat továbbhaladási iránya egy elágazásnál. Hasonlóképpen a programba beiktatott jelzők annak meghatározására szolgálhatnak, hogy a program mely részét akarjuk végrehajtani.

Kézi jelzőművelet



Valamely program feldolgozása során a jelzőket kézzel billentyűzéssel iktathatjuk be a programművelet ellenőrzésére. Van pl. a számológépben olyan költségyszámláló program, amellyel jóváírást és terhelést is feldolgozhatunk. Mivel a terheléseket másképpen kell kezelni, mint a jóváírásokat, beiktatunk egy jelzőt és a programot úgy írjuk meg, hogy a jelző megvizsgálja a bemenő adatokat és ennek megfelelően dolgozza fel.

A fenti vasúti példa következő módosításán meglátjuk, hogyan tudják a vonatok a jelzőművet maguk működtetni. Bemutatjuk azt az elvet, amely szerint a programfeltételek iktatják be a jelzőket. Tételezzük fel, hogy a gyorsvonatokat automatikusan kívánjuk rendeltetési helyükre irányítani.



A vezérlőrendszer minden vonatonál felteszi a kérdést: ez a Debrecen felé menő expressz? Ha a válasz igen, a jelzőmű felemelkedik. Ha a válasz nem, a jelzőmű alsó állásba kerül. Ezt a jelzőművet (flag) kell a későbbi vezérlés során megvizsgálni; ha felső helyzetében van, a vonat Debrecen felé, ha alsó helyzetben van, akkor Szeged felé halad. Hasonlóan kérdezhetjük vissza a programban az utoljára kiszámított érték tulajdonságait: negatív szám? nagyobb 1-nél? és még sok másra kérdezhetünk. Ha a kérdésre adott válasz igen, akkor bekapcsolódik egy jelző, amelyet később vizsgálunk meg, amikor a két lehetőség között választani kell.

A jelzők harmadik alkalmazási módja biztosítja a program számára annak utólagos megállapítását, hogy milyen lépéseken át jutottunk el egy adott pontig. Ez az ellenőrzés olyan ponton szükséges, ahol a további feldolgozás az addigi programlefutás függvénye. Még emlékszünk arra, hogy a programmutató csak saját pillanatnyi állását ismeri, de utólag nem tudja megállapítani, melyik úton jutott ebbe a helyzetbe. Némely esetben szükség van erre az emlékezőképességre, és a programjelzők módját is adnak erre. Programozzunk az egyik feldolgozási ágon 2nd St flg y utasítást, a másikon INV 2nd St flg y utasítást! Nem kívánatos ezt a másik utat szabadon hagyni, mert a későbbi programvégrehajtás hibát okozhat, ha a jelző nincs vissza-lejtetve. Később a 2nd If flg y utasítással könnyen megállapíthatjuk, melyik feldolgozási utat követtük. Egy jelző például megadhatja, hogy a két lehetséges kamatláb közül a programban melyiket használtuk, vagy hogy egy szám előjelét meg kell-e változtatni, mielőtt azt feldolgoznánk stb. A választási lehetőségek egész sorát említhetnénk.

A jelzők különleges rendeltetései

Egyes jelzőket a készüléken belül a következő különleges feladatok végrehajtására programozták:

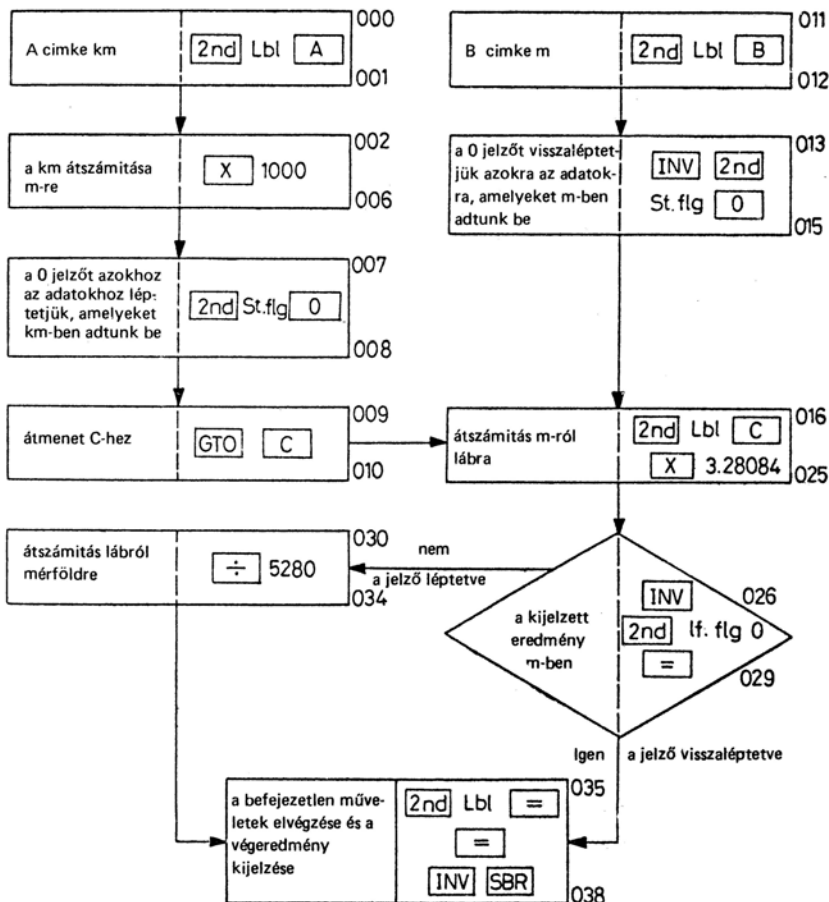
7. jelző: 2nd Op 18 arra utasítja a készüléket, hogy a 7. jelzőt léptesse, ha nincs hibafeltétel. Ha viszont fennáll a hibafeltétel, akkor a 7. jelzőt a 2nd Op 19 utasítással léptetjük (lásd a Különleges vezérlési műveletek című fejezetet).

8. jelző: ennek beiktatásakor a készülék leállítja a programot. Akkor használjuk, ha a futás során hiba történt.
9. jelző: ha a készülékkel együtt nyomtatót használunk, ekkor ezzel a jelzővel a nyomtató párhuzamos üzemét vérelelhetjük. Ha a 9. jelzőt beiktatjuk, a nyomtató párhuzamos üzeme kapcsol, és minden új művelet után kinyomtatja a számítási eredményeket. Ha a 9. jelzőt kiiktatjuk, a gép az eredményt csak nyomtatási utasításra nyomtatja. Nyomtató nélkül a 9. jelző a szokásos módon alkalmazható.

Metrikus egységek átszámítása

Készítsünk olyan programot, mely a métert lábra, a kilométert mérföldre számítja át! Ennek a feladatnak természetesen csékely számú megoldási módja van. A következő módszer szerint a bemeneti adatokat először lábra számítjuk át, majd megvizsgáljuk, hogy a bemeneti adatok mértékegysége km vagy m volt. Ha a vizsgálat azt mutatja, hogy a bemeneti értékeket km-ben fejeztük ki, az eredményt át kell számítani mérföldre, ha a bemenet nem km-ben volt kifejezve, a kijelzett eredményt lábban értendő. A vizsgálat végrehajtása során a gép a közbenső eredményt az R₁-ben tárolja. Átszámítási tényezők:

1 km = 1 000 m, 1 m = 3,28084 láb és 1 mérföld = 5 280 láb.



PROGRAM-UTASÍTÁS				
Lépés	Eljárás	Beírás	Billentyű	Kijelzés
1.	a programtároló törlése és a programmutató visszaállítása		2nd CP	
2.	átállás tanulási üzemmódra		LRN	000 00
3.	a metrikus átszámítási program beírása		LRN	
4.	a programbeírás befejezése		A	mérföld
5.	a kilométerek száma vagy a méterek számának beírása, az eredmény kiszámítása	km m	B	láb

Metrikus átszámítási program

Tárolási hely és billentyűkód	Billentyűzés	Tárolási hely és billentyűkód	Billentyűzés
000 76	2nd Lbl	020 93	.
001 11	A	021 02	2
002 65	X	022 08	8
003 01	1	023 00	0
004 00	0	024 08	8
005 00	0	025 04	4
006 00	0	026 22	INV
007 86	2nd St flg	027 87	2nd If flg
008 00	0	028 00	0
009 61	GTO	029 95	=
010 13	C	030 55	÷
011 76	2nd Lbl	031 05	5
012 12	B	032 02	2
013 22	INV	033 08	8
014 86	2nd St flg	034 00	0
015 00	0	035 76	2nd Lbl
016 76	2nd Lbl	036 95	=
017 13	C	037 95	=
018 65	X	038 92	INV SBR
019 03	3		

Példaképpen írjuk be az előző programot és számítsuk át az 50 m-t lábra, valamint a 90 km-t mérföldre!

Beírás	Billentyű	Kijelzés	Megjegyzések
50	B	164.042	méter → láb
90	A	55.92340909	km → mérföld

Adattároló-elágazások Dsz

Ez a nagy teljesítőképességű utasítás a 0...9 adattárolók tartalmát használja fel annak eldöntésére, hogy egy elágazási átmenet létrejön-e vagy sem. A Dsz-utasítást elsősorban a feltételes hurok képzésére használjuk. A részletes magyarázatot később adjuk meg.

PROGRAMHUROK KÉSZÍTÉSE

Feladatok megoldásakor gyakran szükség lehet olyan utasítássorra, amelyet a kívánt eredmény elérésére megszakítás nélkül többször meg kell ismételni. Ilyen esetekben ún. hurkot programozhatunk. A hurokképzés olyan programozási technika, melynek során a számológépet egy utasítássor folytonos ismétlésére kényszerítjük, a kívánt feladat megoldásáig. Hurok programozása céljából építsünk be a programba olyan utasítást, mely a programmutatót ismételtelen egy korábbi tárolási helyre állítja be.

Feltétel nélküli hurokképzés

A feltétel nélküli hurok képzésre két lehetőség kínálkozik:

1. RST utasítással állandóan visszaállítjuk a programot a 000 tárolóhelyre,
2. GTO utasítással állandóan visszaállítjuk a programot egy megadott címre.

Készítsünk olyan programot, mely négyesével számol! Az egyszerű

$$+ 4 = 2nd \text{ Pause RST}$$

utasítás teljesíti ezt a feladatot, ha azt a program.tároló bemenetére adjuk. A fenti utasítás beadása után oldjuk fel a programozási üzemmódot, menjünk vissza a 000 tárolóhelyre, adjunk be egy indítószámot, nyomjuk meg az R/S billentyűt, és figyeljük meg a számlálást! Ha az utasítást a 020 tárolóhelytől kezdve akarjuk a programtárolóba bevinni, akkor ugyanezen hatás elérésére az RST-t a GTO 020 utasítással helyettesíthetjük. Csak ügyeljünk rá, hogy induláskor a végrehajtást a 020 tárolóhelynél kell kezdenünk!

Óvatosan bánjunk az RST utasítással, mert az összes jelzőt kiiktatja, és az alprogram-visszaléptető tárolót is törli!

Tárolási hely és billentyűkód	Billentyűzés	Megjegyzések
020 25	CLR	a kijelzés törlése
021 76	2nd Lbl	
022 85	+	ezt a szakaszt + címkével jelöljük meg
023 85	+	
024 04	4	
025 95	=	
026 66	2nd Pause	minden egyes számlálás kijelzése
027 61	GTO	vissza a + címkéhez
028 85	+	

Feltételhez kötött hurokképzés

Az előző számlálási folyamatot feltételhez kötött elágazási utasítással is hiánytalanul vezérelhetjük. A szám-

lálás haladjon ismét 0-tól 20-ig!

Tárolási hely és

billentyűkód

Billentyűzés

Megjegyzések

000 02

2

001 00

0

002 32

$x \neq t$

003 25

CLR

004 76

2nd Lbl

programrész jelölése A címkével

005 11

A

006 85

+

007 04

4

008 95

=

009 66

2nd Pause

010 22

INV

011 77

2nd $x \geq t$

a vizsgálat megfordítása és elágazás A-hoz, ha

012 11

A

az utolsó számérték 20

013 91

R/S

leállás, ha a 20-at elértük.

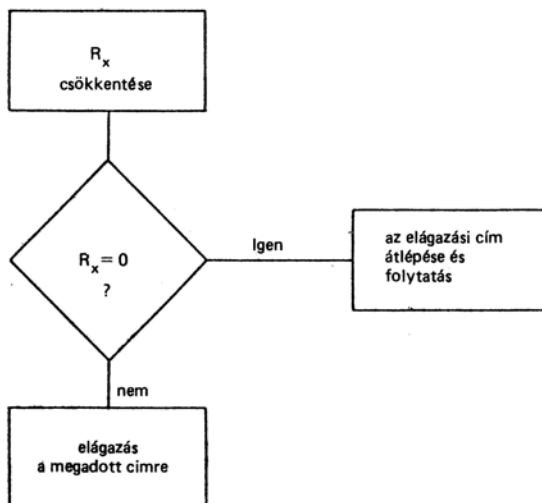
Itt a hurkot az INV $x \geq t$ 2nd utasítással vezéreltük.

Hurokképzés a feltételhez kötött elágaztatási utasítással Dsz

Ha tudjuk, hogy egy utasítást milyen gyakran kell ismételni, akkor hurokképzéshez használhatjuk a Dsz utasítást (decrement and skip on zero). Az utasítás 2nd Dsz X, melyet egy elágazási cím követ.

X a 10 adattároló (0...9) közül annak a száma, amelyet az utasítás céljára lefoglalunk.

Ez a sokoldalú utasítás az x adattároló tartalmának nagyságát 1-gyel csökkenti (az 1-nél kisebb értékű adattárolót 0-ra emeli) és ezután megvizsgálja az x tároló tartalmát. (A következő magyarázatban az $R_x =$ az x adattároló tartalma jelölést használjuk). Ha $R_x = 0$, az elágazási címet átugratjuk. Egyéb esetekben a feldolgozási folyamat a megadott címen folytatódik. A Dsz utasítás csökkenti az x tárolót és 0-nál átlépi az elágazást. A következő ábra szemlélteti a folyamatot.



A többi elágaztatási utasításhoz hasonlóan a Dsz utasítást is beadhatjuk billentyűzve; de lehet egy program része is. Billentyűzzük be a következőket, és figyeljük meg mi történik!

Billentyű	Kijelzés	Megjegyzés
2nd CP	0.	a programtároló törlése
2 STO 6	2.	a 2-t a 06 adattárolóban tároljuk
2nd Dsz 6 136	2.	az R_6 csökkentése 1-gyel, majd kérdés: $R_6 = 0$? (ha nem, akkor elágazás)
LRN	136 00	az átmenet 136-ra megtörtént
LRN RCL 06	1.	R_6 2 volt: most a Dsz miatt 1 lesz
2nd Dsz 6 111	1.	csökkentés és újabb vizsgálat
LRN	136 00	nincs átmenet, mert $R_6 = 0$
LRN RCL 06	0.	R_6 valóban 0.

A Dsz nem más, mint egy számlálórendszer, amely visszaszámlálásnál hurkot képez 0-ig, majd utána más utasításra tér át.

A programban fellépő előnyoinek bemutatására még egyszer utalunk a számlálási példára. Láthatjuk, hogy ha 20-ig négyes lépésekben számolunk, (+4) a huroknak ötször kell átfutnia.

Tárolási hely és

billentyűkód	Billentyűzés	Megjegyzés
000 47	2nd CMs	az összes adattároló törlése
001 05	5	
002 48	2nd Exc	az 5-t a 00 tárolóban tároljuk és a kijelzést töröljük
003 00	0	
004 76	2nd Lbl	ezt a részt az A címkevel jelöljük meg
005 11	A	
006 85	+	
007 04	4	
008 95	=	
009 66	2nd Pause	minden egyes számlálás kijelzése
010 97	2nd Dsz	a 00 csökkentése és annak vizsgálata, hogy $R_0 = 0$ -e
011 00	0	
012 11	A	$R_0 = 0$ esetben átmenet az A címke
013 91	R/S	állj, ha $R_0 = 0$.

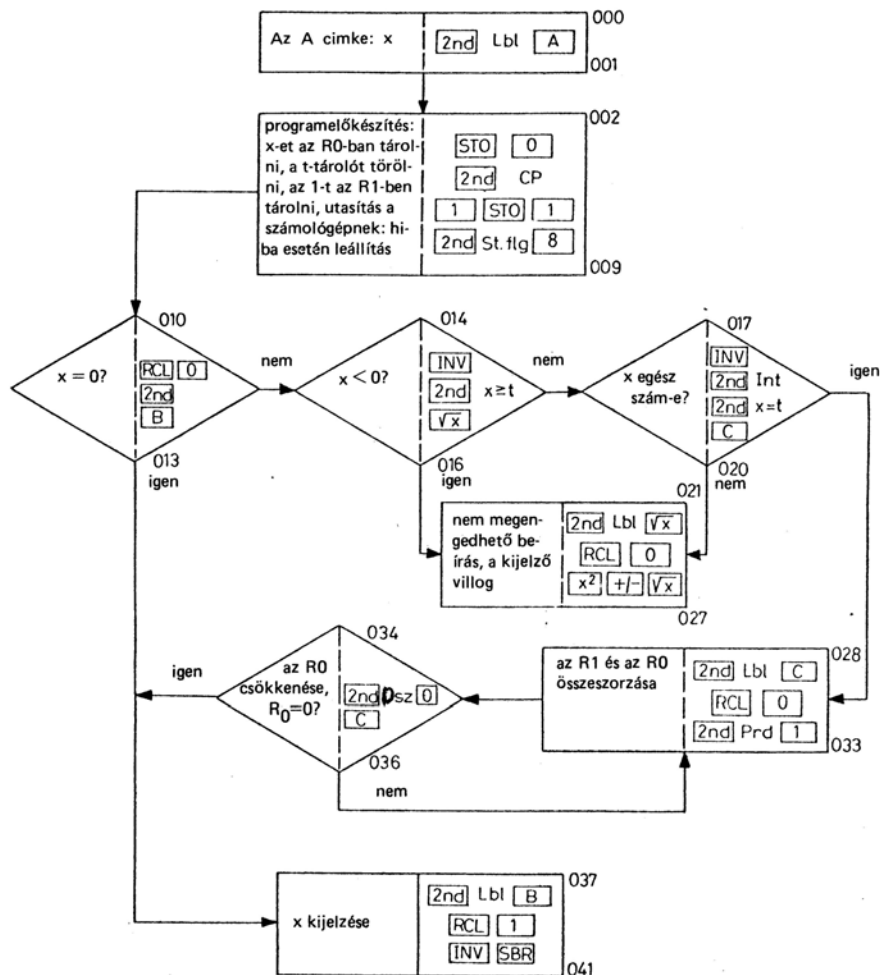
Hasonlóképpen, a Dsz utasítás a számegyenes negatív oldaláról kezdve is növelni tudja az értékét, mindig 1-t adva az R_x -hez. A fenti példában alkalmazhattuk volna -5-t is. Hasonló módon növel vagy csökkent az INV 2nd Dsz utasítás is, de az elágazási címet most nem a 0-nál, hanem a „nem nullánál” ugorjuk át.

Ezt a témát részletesen az V. fejezet a Csökkentés és átmenet 0-nál c. részben tárgyaljuk.

Ez az utasítás akkor is előnyös, ha 1-től N-ig haladó számsorral kell számolnunk. Ezt a számsort Dsz utasítással, hurokképzéssel lehet feldolgozni. Ennek során a kifejezést különböző változókra számítjuk ki és a gépet arra utasítjuk, hogy az adattároló tartalmát hívja be és minden alkalmazáskor a változó értékét eggyel csökkenti. Ügyeljünk arra, hogy a sor a valóságban N-től 1-ig tart, mivel a Dsz utasítás csökkent!

Az x! program

A Dsz-hurok alkalmazásának gyakorlására írjunk programot az x faktoriális, számítására $x! = x \cdot (x-1) \cdot (x-2) \dots$ amíg a zárójeles kifejezés értéke el nem éri az 1-et. A meghatározás szerint az x-nek pozitív egész számnak kell lennie, továbbá $0! = 1$.



Az x! program táblázata

Tárolóhely és billentyűzés	Billentyűzés	Tárolóhely és billentyűkód	Billentyűzés
000 76	2nd Lbl	021 76	2nd Lbl
001 11	A	022 34	$\sqrt{x}$
002 42	STO	023 43	RCL
003 00	0	024 00	0
004 29	2nd CP	025 33	x^2
005 01	1	026 94	$\pm/\cdot$
006 42	STO	027 34	$\sqrt{x}$
007 01	1	028 76	2nd Lbl
008 86	2nd St flg	029 13	C
009 08	8	030 43	RCL
010 43	RCL	031 00	0
011 00	0	032 49	2nd Prd
012 67	2nd x=t	033 01	1
013 12	B	034 97	2nd Dsz
014 22	INV	035 00	0
015 77	2nd $x \geq t$	036 13	C
016 34	$\sqrt{x}$	037 76	2nd Lbl
017 22	INV	038 12	B
018 59	2nd Int	039 43	RCL
019 67	2nd x=t	040 01	1
020 13	C	041 92	INV SBR

A mintaprogramban az 1-t az R₁-ben tároltuk, hogy a szorzást a tárolóaritmetikával végezhessük el. A programozási gyakorlat teljessé tételére egybefoglaltuk az első három feltételes elágazást, hogy a nem megengedhető beírásokat kiküszöböljük. Figyeljük meg, hogy nem megengedhető beírás esetén a 027 tárolóhelyen programozott hibafeltétel megszakítja a programot, mert az előző helyre a 8. jelzőt iktattuk be! A tényleges hurok a 028..036 tárolóhelyek közt található.

Ennek a programnak az alkalmazása igen egyszerű. Adjunk be egy 70-nél kisebb x-értéket és nyomjuk meg az A billentyűt! A 70! értéke túllépi a készülék számítási tartományát. Példaként számítsuk ki 6!, -2!, 0!, 7,3!, 39! értékét.

Beadás	Billentyű	Kijelzés	Megjegyzések
6	A	720	6!
2	$\pm/\cdot$ A	" 2 "	nem megengedett beírás
	CLR	0	a hibafeltétel törlése
0	A	1	0!
7.3	A	" 7.3 "	nem megengedett beírás
	CLR	0	a hibafeltétel törlése
39	A	2.0397882 46	39!

Megjegyzés: a kijelzés oszlopában levő idézőjelek a kijelző villogását jelzik.

Másodfokú egyenletek megoldásának programja

A programozási technika egyik legszemléletesebb példája a másodfokú egyenlet megoldásának programozása.

Írjunk programot a következő másodfokú egyenlet valós és komplex gyökeinek meghatározására!

$$ax^2 + bx + c = 0 \quad (a \neq 0)$$

Az x_1 és x_2 gyök a következő képlettel számítható

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Ha a b^2-4ac kifejezés értéke pozitív, vagy 0, a gyökök valósak, és azokat a fenti képletekkel lehet kiszámítani. Ha a b^2-4ac értéke negatív, a gyökök komplexek. A valós és a képzetes részt külön választva:

$$x_1 = \operatorname{Re} x + j \cdot \operatorname{Im} x$$

$$x_2 = \operatorname{Re} x - j \cdot \operatorname{Im} x$$

$$\text{ahol } \operatorname{Re} x = \frac{-b}{2a}$$

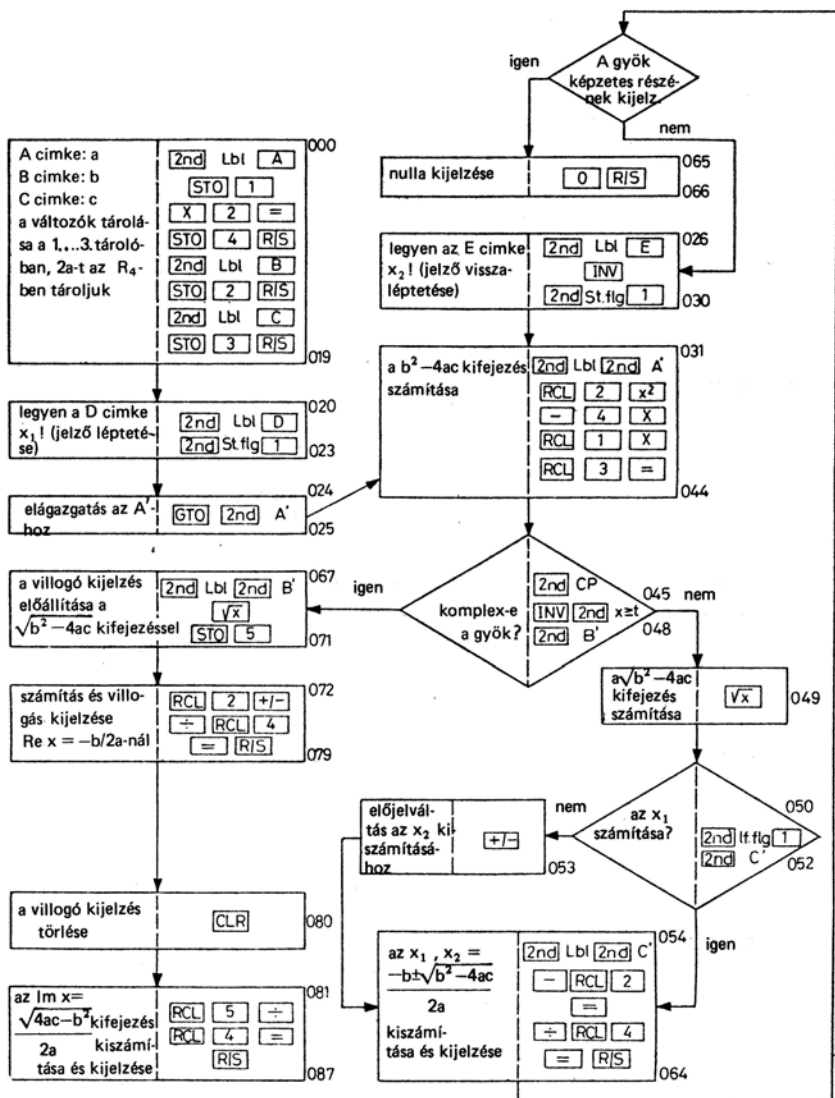
$$j = \sqrt{-1}$$

$$\operatorname{Im} x = \frac{\sqrt{|b^2-4ac|}}{2a}$$

Mivel az x_1 és az x_2 ugyanazzal a képlettel számítható, programlépéseket takaríthatunk meg a következő módon; vonjuk össze ezt a programrészt, és iktassunk be egy jelzőt annak jelölésére, hogy melyik gyököt kell kiszámítani! Különálló programrész szükséges a gyökök valós és képzetes részre való felbontására. Annak kiderítésére, hogy a gyökök komplexek vagy valósak, megvizsgáljuk, hogy a b^2-4ac kifejezés negatív-e. Gondoljunk arra, hogy ha a gyökök komplexek, akkor az x_2 -t nem kell kiszámítanunk, mert a $\operatorname{Re} x$ és az $\operatorname{Im} x$ értéke mindkét gyöknél azonos.

Biztosítani kell annak kijelzését is, hogy a gyökök valósak vagy komplexek. Miután komplex gyök esetén a b^2-4ac negatív, a kijelző villogással jelzi a négyzetgyökvonást. Vegyük figyelembe azt is, hogy ha a b^2-4ac negatív, voltaképpen a $4ac-b^2$ kijelzést számítjuk ki! Ezt az eredményt tárolhatjuk, és később a gyökök képzetes részének meghatározásakor ismét felhasználhatjuk. A következő mintaprogramban a gyökök képzetes részét az R/S billentyűvel, a valós rész kiszámítása után határozzuk meg. Abban az esetben, ha a gyöknek nincs képzetes része, biztonsági okokból a készülék 0-t jelez ki. Ez a program nem alkalmas alprogramnak, mert = és R/S utasítást is tartalmaz.

PROGRAMUTASÍTÁSOK				
Lépés	tevékenység	beírás	billentyű	kijelzés
1.	a programtároló törlése és a programmutató visszaállítása		2nd CP	0.
2.	átállítás tanulási üzemmódba		LRN	000 00
3.	programbeadás: másodfokú egyenlet			
4.	a tanulási üzemmód feloldása		LRN	0
5.	az a beírása a=0	a	A	a
6.	a b beírása	b	B	b
7.	a c beírása	c	C	c
8.	az x_1 kiszámítása: ha a kijelző villog, a gyöknek képzetes része is van		D	x_1 (valós rész)
	a képzetes rész kiszámítása		R/S	x_1 (képzetes rész)
9.	ha villog a kijelző, a gyök komplex		E	x_2 (valós rész)
	a képzetes rész kiszámítása		R/S	x_2 (képzetes rész)



Program másodfokú egyenlet megoldásához

Tárolóhely és billentyűkód	Billentyűzés	Tárolóhely és billentyűkód	Billentyűzés	Tárolóhely és billentyűkód	Billentyűzés
000 76	2nd Lbl	030 01	1	059 95	=
001 11	A	031 76	2nd Lbl	060 55	÷
002 42	STO	032 16	2nd A'	061 43	RCL
003 01	1	033 43	RCL	062 04	4
004 65	X	034 02	2	063 95	=
005 02	2	035 33	x^2	064 91	R/S
006 95	=	036 75	—	065 00	0
007 42	STO	037 04	4	066 91	R/S
008 04	4	038 65	X	067 76	2nd Lbl
009 91	R/S	039 43	RCL	068 17	2nd B'
010 76	2nd Lbl	040 01	1	069 34	$\sqrt{x}$
011 12	B	041 65	X	070 42	STO
012 42	STO	042 43	RCL	071 05	5
013 02	2	043 03	3	072 43	RCL
014 91	R/S	044 95	=	073 02	2
015 76	2nd Lbl	045 29	2nd CP	074 94	+/-
016 13	C	046 22	INV	075 55	÷
017 42	STO	047 77	2nd $x \geq t$	076 43	RCL
018 03	3	048 17	2nd B'	077 04	4
019 91	R/S	049 34	$\sqrt{x}$	078 95	=
020 76	2nd Lbl	050 87	2nd If flg	079 91	R/S
021 14	D	051 01	1	080 25	CLR
022 86	2nd St flg	052 18	2nd C'	081 43	RCL
023 01	1	053 94	+/-	082 05	5
024 61	GTO	054 76	2nd Lbl	083 55	÷
025 16	2nd A'	055 18	2nd C'	084 43	RCL
026 76	2nd Lbl	056 75	—	085 04	4
027 15	E	057 43	RCL	086 95	=
028 22	INV	058 02	2	087 91	R/S
029 86	2nd St flg				

Példaképpen számítsuk ki a következő egyenlet gyökeit!

$$1,5x^2 + 3,7x + 2,25 = 0$$

Beírás	Billentyű	Kijelzés	Megjegyzés
1.5	A	3.	a → 2a
3.7	B	3.7	b
2.25	C	2.25	c
	D	-1.088036702	az x_1 számítása (a nem villogó kijelzés a valós gyökre utal)
	E	-1.378629965	az x_2 számítása

Számítsuk ki a következő egyenlet gyökeit! $x^2 + 2x + 17 = 0$.

Beírás	Billentyű	Kijelzés	Megjegyzés
1	A	2.	a → 2a
2	B	2.	b
17	C	17.	c
	D	"-1."	a Re x kijelzése; a villogó kijelzés komplex gyököt jelez;
	R/S	4.	az Imx kijelzése

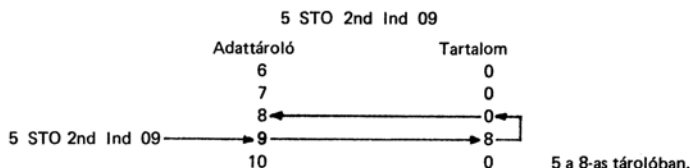
Közvetett utasítások programozása

A 2nd Ind közvetett utasítást felhasználhatjuk az adattároló műveletek, az elágazások, a különleges vezérlési műveletek és a software-programcímzés bővítésére. Az elv egyszerű. Egy adattárolót nem arra használunk, hogy a szükséges információt megtaláljuk, hanem azért, hogy megtudjuk, ez az információ hol van. Ez hasonló ahhoz, mintha valakit megbíznánk: „menj és kérdezd meg X-et, hogy hol van Y!"; ahelyett, hogy ezt kérnénk: „menj és keresd meg Y-t!” Nyilvánvaló, hogy ha X tudja, hogy hol van Y, a kérdező is azonnal tudja Y pillanatnyi tartózkodási helyét. Ezzel szemben órákig eltarthat, míg valaki felkutatja és megtalálja X-et.

A programozás területén néha lényegesen egyszerűbb közvetett úton informálódni. Adódhat olyan helyzet, hogy Y közvetlenül nem található meg, ilyenkor a közvetett módszer az egyetlen lehetőség. A közvetett utasítások elé 2nd Ind-et kell billentyűzni, és ezt egy adattároló számának kell követnie. Ez az adattároló tartalmazza a szükséges információt az utasítás végrehajtásához.

Közvetett művelet az adattárolóval

Valamennyi adattároló utasításnál (tárolás, behívás, tároló-csere, tároló-összeadás, tároló-szorzás) használható a közvetett címzés. Tekintsük a következőket:



Írjunk most egy programszakaszt több adattároló törlésére! Egyszerűsítés céljából töröljük az 1-es tárolót x segítségével, ahol x változtatható szám!

Tárolóhely billentyűkód	Billentyűzés	Megjegyzés
000 76	2nd Lbl	az x beírása és az A billentyűzése
001 11	A	
002 42	STO	az x a 00 adattárolóban
003 00	0	
004 76	2nd Lbl	
005 12	B	
006 25	CLR	
007 72	STO 2nd Ind	nullát kell abban a tárolóban tárolni, amely a 00-val van megadva
008 00	0	
009 97	2nd Dsz	Dsz hurok a 00 tároló számára
010 00	0	
011 12	B	átmenet a B-hez, ha R00 ≠ 0
012 92	INV SBR	a program leállítása, ha a 00-tároló elérte a nullát.

A hurok első átfutáskor x a 00 tárolóban van, s így az x tárolóban a CLR STO 2nd Ind 00 utasítás egy nullát tárol. A Dsz csökkenti a 00 tároló tartalmát (x-1)-re. A közvetett tárolási utasítással a nullát most már az X-1 tárolóban tároljuk stb. A tárolókat fordított sorrendben állítottuk nullára, ami azonban nem zavaró. Tudunk-e olyan programot írni, amely segítségével azokat a számok sorrendjében törölhetjük?

Ügyeljünk a különleges billentyűkódra, a 72-re is, amely az STO 2nd Ind utasításhoz tartozik! Célzerű itt is több közvetett utasítást összevonni, így programtároló helyeket takarítunk meg. A teljes felsorolást az V. rész „Utasításkódok” című fejezete tartalmazza.

Közvetett elágazási utasítások

A közvetett címzések célszerűen kiterjeszthetők programelágazásokra is. Még emlékezünk arra, hogy egy elágazási cím megadására két lehetőség van; az abszolút cím vagy a programtárolóban levő címke. A közvetett címzés más, rugalmas módszert ad. Megadjuk azt az adattárolót, amelyben a kívánt abszolút cím megtalálható. A címzést nem lehet adattárolóban tárolni.

Közvetett elágaztatás esetén a feltétel nélküli GTO, SBR vagy feltételhez kötött 2nd x=t; 2nd Dsz stb. elágazási utasítások után be kell iktatni a 2nd Ind utasítást. Ezt az utasítást annak az adattárolónak a címével kell lezárni, amely a programtárolóhely abszolút címét tartalmazza; ez utóbbihoz kívánunk elágazást létesíteni. Próbáljuk ki az utasítást a billentyűzeten!

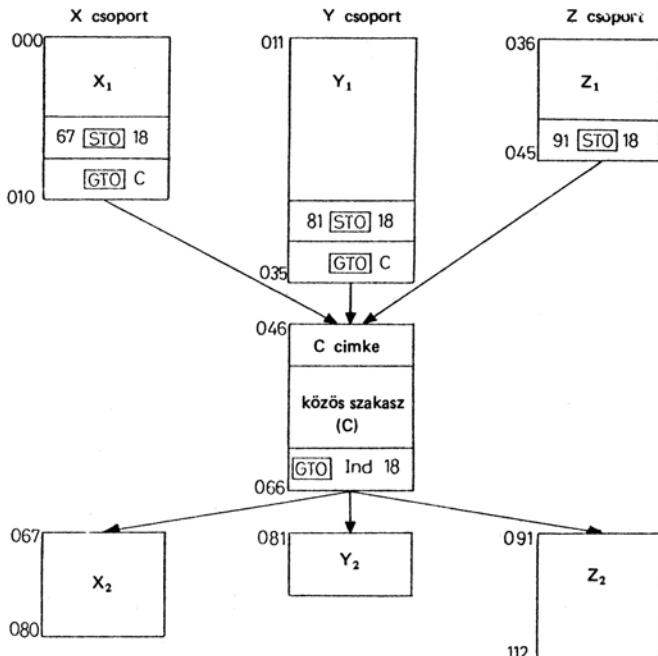
Billentyűzés	Kijelzés	Megjegyzés
35 STO 18	35.	35 a 18-as adattárolóban
GTO 2nd Ind 18 LRN	035 00	a programmutatót a 035 tárolóhelyre állítottuk be

A Dsz utasítás közvetett úton is megadhatja azt a tárolót, amelynek tartalmát csökkenteni kell. Ilyen utasítás alkalmazásakor 2nd Dsz 2nd Ind 12 2nd Ind 14 billentyűzés használható. Mint a Dsz utasításnál használt adattároló, az elágazási cím is elérhető közvetett módon.

A következő ábrázolás érthetővé teszi ezt az elágazási módszert, és alkalmazását. Tegyük fel, hogy az alább megadott három utasításcsoportot be kell iktatni egy programba!

X csoport	Y csoport	Z csoport
X_1	Y_1	Z_1
C	C	C
X_2	Y_2	Z_2

A középső rész (C) minden utasításcsoportban ugyanaz, ezért kézenfekvő ennek a résznek csupán egyszeri megírása. Egyszerű dolog lenne az X_1 és Y_1 szakasz végén a C-hez való eljutás érdekében GTO utasítást alkalmazni. (Z_1 közvetlenül C-hez megy). Kérdés azonban, hogy ágazik el a program X_2 -höz, Y_2 -höz és Z_2 -höz. A feladat közvetett címzéssel oldható meg. Tároljuk egyszerűen a harmadik rész címeit a C-hez való elágazás előtt, zárjuk le a C-t a GTO 2nd Ind utasítással! A magyarázat kedvéért az egyes szakaszok elején és végén a tárolóhelyeket önkényesen választottuk ki.



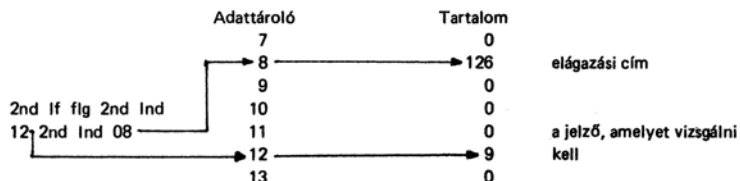
Egyéb tulajdonságok

A 2nd Ind utasítással a programjelzők közvetett módon be- és visszaléptethetők. A rögzített tizedesvessző-beállítást is ellenőrizni lehet ezzel az utasítással.

Hasonlóképpen lehetséges a közvetett jelző vezérlése úgy, hogy a jelzőszámot egy adattárolóban tároljuk.

Pl. tároljuk a 6-ot az R12-ben, és az utasítássort zárjuk a 2nd St flg 2nd Ind 12 utasítással! Ekkor valójában a 6-os jelzőt iktattuk be, miközben a 2nd If flg 2nd Ind 12 A a 6-os jelző állapotától függően az A címkehez ágaztat el.

Minthogy a 2nd If flg egy elágazási utasítás, az elágazás közvetett módon is végrehajtható. Ezért a 2nd If flg, a 2nd Ind 12 és a 2nd Ind 08 is megfelelő.



Egyenértékű utasítás a 2nd If flg 9 126. Írjuk be ezt a példát a készülékbe és figyeljük meg annak hatását!

Billejtő (Instruction)	Kijelzés (Output)	Megjegyzés (Note)
126 STO 08	126.	a 126 a 08 tárolóban (126 in register 08)
9 STO 12	9.	9 a 12 tárolóban (9 in register 12)
2nd St flg 9	9.	a 9-es jelző beléptetése (setting the 9th flag)
2nd If flg 2nd Ind 12		
2nd Ind 8 LRN	126 00	elágazás a 126 tárolóhelyre. (branch to memory location 126)

Hasonlóan lehet a tizedesvessző-beállítást közvetve vezérelni, amint ezt a következő példa is mutatja.

Billentyű	Kijelezés	Megjegyzés
2 STO 12	2.	a 2 a 12 tárolóban
2nd Fix 2nd Ind 12	2.00	a készülék tizedesvesszőjének beállítása a 2. tizedes helyre

A közvetett címzésről részletesebben az V. részben írunk.

A program optimalizálása (A programlépések csökkentése)

Egy program optimalizálásánál két fő szempontot követünk. Egyrészt a program alkalmazását egyszerűsítjük, másrészt a programot olyan tömörre tehetjük, hogy ezzel a tárolótartomány elosztása kedvezőbbé váljon.

Programozási technikák az egyszerűsítés érdekében

Általában a jól megírt program kevés billentyűzéssel, akadálytalanul végrehajtható.

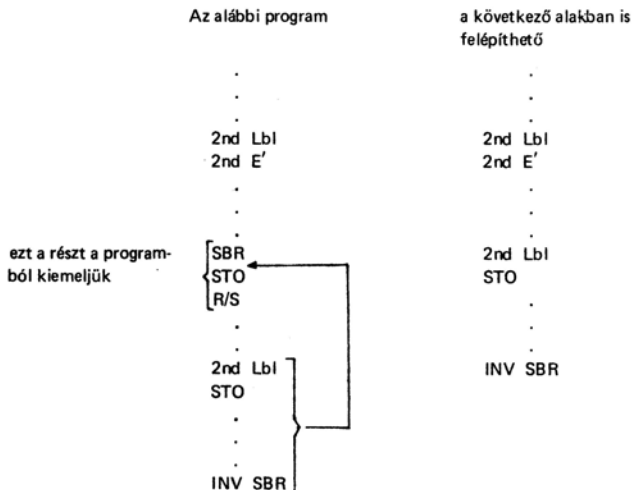
Némely programnál helytelen beírás vagy hibás billentyűzési utasítás miatt a teljes megoldást újra kell kezdeni, ami sokszor terhes és időrabló munka, különösen a bonyolult és hosszú programoknál. A hibakeresés és javítás egyszerűsítése fontos a program alkalmazásának megkönnyítése szempontjából is. Ajánlatos az eredeti, kiindulási adatokat megőrizni, ill. tárolni. Ajánlatos ezenkívül a tárolóaritmetika által végrehajtott programokat STO utasítással kezdeni, hogy az adattárolók törlése nélkül ismétlődő legyen.

Programozási módszerek a lépésszám csökkentésére

A program tömörre tétele, a lépésszám csökkentése sok időt vesz igénybe. Ha a program a programtárolók kapacitásával összhangban van, és helyesen működik, akkor minden további tömörítés felesleges időpazarlás.

A programlépések számának csökkentésekor figyelmünket először az utasításokra kell fordítanunk, amelyek egyénél többször fordulnak elő. Ha ezek az utasítások elég hosszúak és elég gyakoriak, akkor alprogramokkal is helyettesíthetjük azokat. Így módon programtároló helyet szabadíthatunk fel. A sok alprogrammal rendelkező programok esetén is túlléphetjük a programtároló-kapacitást. Ilyenkor van nagy jelentősége az alprogramok optimalizálásának.

Sokféle módszer van arra, hogy lehet különálló programrészek kombinálásával tárolóhelyeket megtakarítani. Ha pl. egy alprogram behívása egy másik programszakasz utolsó művelete, akkor az alprogramot egy szintre hozzátjuk az előzővel.



Az előbbi példánál nemcsak a programlépések számát csökkentettük, hanem az alprogram-vissaléptető tároló egyik mezőjét is felszabadítottuk. Az INV SBR úgy működik most, mint egy R/S utasítás, mert az alprogram-vissaléptető tároló nincs lefoglalva. Következő példaként vizsgáljuk meg a következő két utasítást!

működőképes program-részlet	észszerű program-részlet
.	.
.	.
2nd x=t	2nd x=t
D	D
.	.
1	2nd Lbl
STO	D
4	1
INV SBR	STO
2nd Lbl	4
D	INV SBR
1	
STO	
4	
INV SBR	

Itt a vizsgálati eredményektől függően a 0,1-et vagy 1-et kell tárolni. Mindkét szakasznak ugyanaz a szerepe, a második azonban 4 lépéssel rövidebb, mert a nem hozzá tartozó utasításokat (az ábrán a bekeretezett részt) elhagytuk.

Különálló programrészek összevonása is igen előnyös. A következő példában a további számításokhoz a kijelzett számnak csak két számjegyre kerekített értékét akarjuk felhasználni. Jelen esetben nem megoldás az, ha a készüléket két számjegyre kapcsoljuk, mert így a továbbra is kerekítetlen számértékkel számol a gép.

működőképes program-részlet	észszerű program-részlet
.	.
.	.
.	.
X	2nd Fix
1	2
0	EE
0	INV
=	EE
2nd Int	2nd Fix
÷	9
1	.
0	.
0	.
=	
.	
.	
.	

A baloldali programrész célja és módszere az előző példában világos és közvetlen. A második eljárás ésszerűbb, de nem olyan könnyen érthető. Mivel az EE utasítás csak a kijelzett értékekre vonatkozik, a nem kívánt helyeket a rögzített tizedesvessző beállítás után törli. A programrész ezenkívül a feldolgozást a kerekített értékkel folytatja.

A következő programrészletek három módszert mutatnak, ugyanannak a műveletnek végrehajtására; 10 000-et adunk hozzá a kijelzőtároló tartalmához.

.	.	.
.	.	.
.	.	.
+	+	+
1	1	4
0	EE	INV
0	4	2nd log
0	=	=
=	.	.
.	.	.
.	.	.

A 2. és 3. programszakasz ugyanannyi programtárolóhelyet foglal el. A második módszer csak akkor előnyös, ha a hatványkitevős kijelzési módot meg akarjuk tartani.

A számológép teljesítőképeségének fokozatos megismerésével magunk is fogunk alkalmas rövidített eljárásokat találni. Jegyezzük meg azokat a jövőbeni alkalmazások céljára. Programozási feladataink lerövidítése érdekében! Addig azonban programoptimalizálásra használjuk fel a készülék előnyös tulajdonságait, amelyekkel a programlépések száma csökkenthető. Ilyenek pl. a SUM és a 2nd Prd tárolóművelet, a közvetett utasítások.

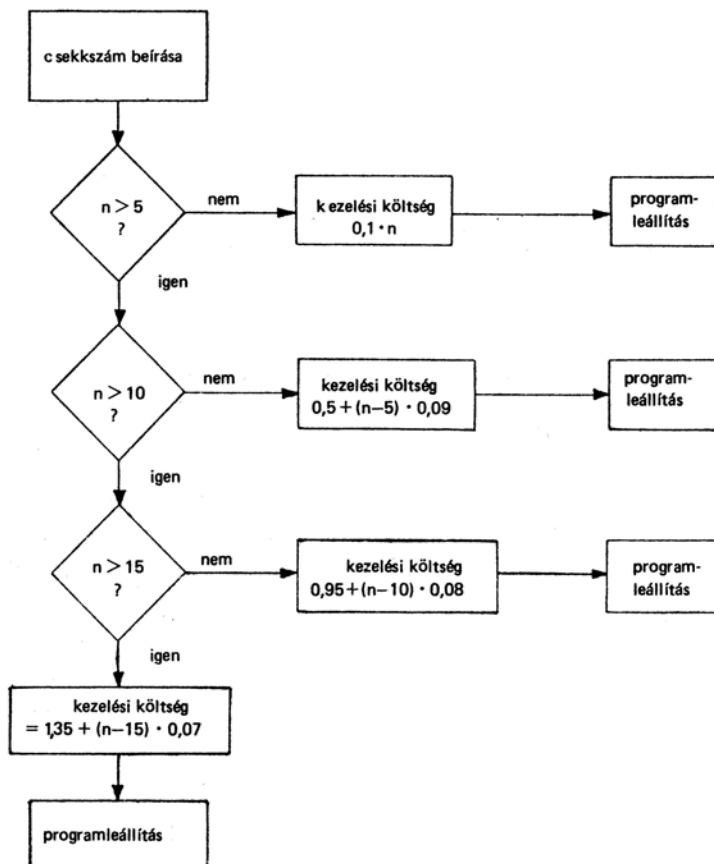
Ha még így is nehézségekbe ütközik a program elhelyezése a kijelölt tárolótérben, szükségessé válhat a program szakaszokra bontása és közbeni eredmények kiszámítása, még mielőtt a végeredmény számítását programozzuk. A közvetlen formában történő programozásra van egy másik lehetőség is, amelyet a következő példán mutatunk be.

Program a kezelési költség számításához

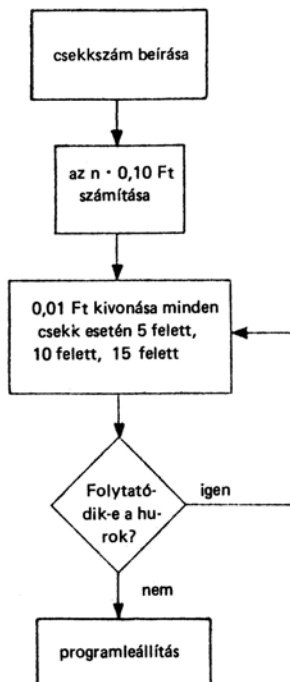
A következőkben egy módszert ismertetünk, amellyel pl. egy bank kiszámíthatja ügyfeleinek a számlák kezeléséért járó havi kezelési költségét. A kezelési költséget pl. az alábbi díjszabás szerint számítják:

0,10 Ft csekkenként az első 5 csekkre
 0,09 Ft csekkenként a következő 5 csekkre (6 ... 10)
 0,08 Ft csekkenként a következő 5 csekkre (11 ... 15)
 0,07 Ft csekkenként minden további csekkre 15 felett.

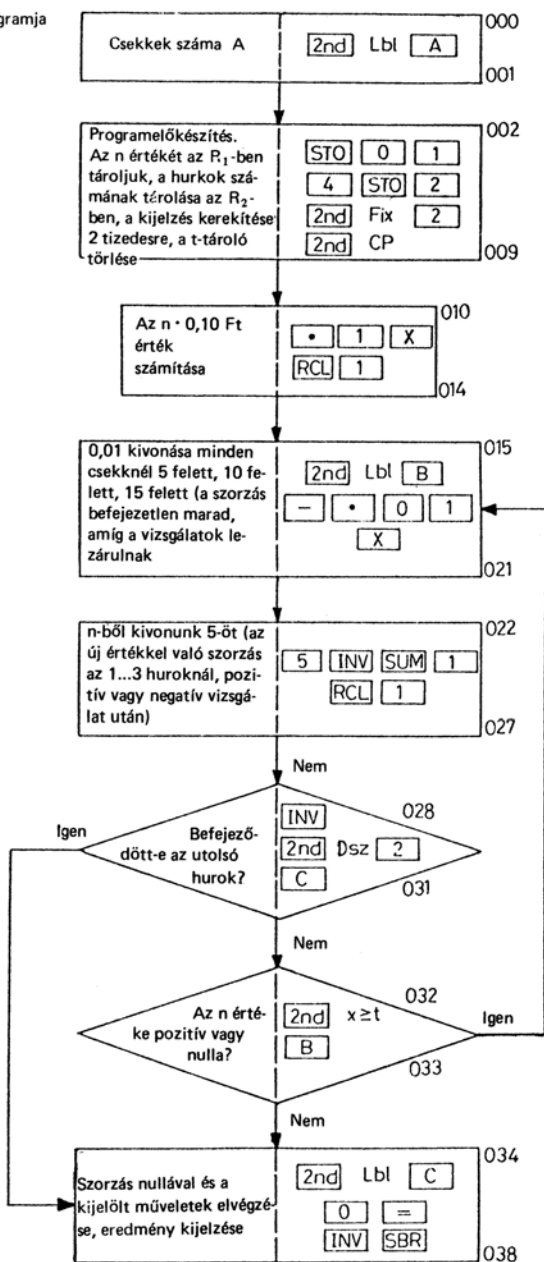
Közvetlen megoldást mutat a következő folyamatábra.



Ha a programot a vázolt megoldási módszer szerint készítjük, legalább 80 vagy 90 tárolóhelyre van szükségünk. Bár egy ilyen program beírása nem jelent gondot, előfordulhat, hogy más programrészek számára is kell helyet biztosítani, és ezt a programot lényegesen rövidebbé kell tenni. Feltehetően más eljárással kevesebb programlépésre lenne szükség. Tekintsük a következő megoldást!



Első pillanatban úgy tűnik, hogy a vázolt gondolatmenet alapján készített program könnyen bevihető számológépünk programtárolójába. Azonban a fenti gondolatmenet érvelése nem világos első olvasásra. Vizsgáljuk meg röviden ezt a logikát!



A program a 022 tárolóhelyig folyamatos, a 021 lépésnél a szorzás befejezetlen marad, amíg az n értékének helyesbítése és a vizsgálatok befejezése meg nem történik. Hurkot alkalmazunk, hogy a kezelési illeket minden 5 feletti cseknél 0,09 Ft-tal, minden 10 felettinél 0,08 Ft-tal, és minden 15 felettinél 0,07 Ft-tal csökkentünk. A 2nd Dsz utasítással megkérdezzük, hogy melyik hurok fut éppen át. Az 1..3 huroknál a gép megvizsgálja az n értékét; ha az negatív, akkor egy nullát visz be a kijelzőbe, és elvégzi a kijelölt szorzást. A programot a teljes kezelési költség kiszámítása után lezárjuk.

Ha eljutottunk a 4. hurokhoz, a befejezetlen szorzást mindig nullával zárjuk le, mert különben minden 20 feletti csekket 0,06-ra csökkentett értékkel számolnánk. A program ezután kiszámítja a teljes kezelési illeket, és rögzíti azt.

Ez az utolsó hurok nem szükséges a számításához. Ha azonban elhagynánk, további programutasításokra lenne szükség; pedig a cél az volt, hogy a programot egyszerűsítsük.

A kezelési költség számítására eddig két megoldást adtunk. Ha tekintetbe vesszük, hogy egy probléma megoldását sokféleképp programozhatjuk, akkor a fenti két szélsőséges példa a programozási módszerek közötti nagy különbséget mutatja. Természetesen engedményeket kell tenni. Ebben a példában a második módszerhez a felénél is kevesebb tárolóhely szükséges, mégis az első megoldás szerinti program fut le gyorsabban. A megoldás módjától függetlenül mindig gondoljunk arra, hogy a helyes megoldás az, amelyik számunkra a legjobb.

A kezelési költségek számításának programja

Tárolóhely és billentyűkód	Billentyűzés	Tárolóhely és billentyűkód	Billentyűzés
000 76	2nd Lbl	020 01	1
001 11	A	021 65	X
002 42	STO	022 05	5
003 01	0 1	023 22	INV
004 04	4	024 44	SUM
005 42	STO	025 01	1
006 02	2	026 43	RCL
007 58	2nd Fix	027 01	1
008 02	2	028 22	INV
009 29	2nd CP	029 97	2nd Dsz
010 93	.	030 02	2
011 01	1	031 13	C
012 65	X	032 77	2nd $x \geq t$
013 43	RCL	033 12	B
014 01	1	034 76	2nd Lbl
015 76	2nd Lbl	035 13	C
016 12	B	036 00	0
017 75	—	037 95	=
018 93	.	038 92	INV SBR
019 00	0		

A következő program végrehajtása céljából írjuk be egyszerűen a csekkek számát és nyomjuk meg az A billentyűt! Egy csekk pl. 0,10 Ft-ba kerül, 6 csekk 0,59 Ft-ba és 63 csekk ára 4,71 Ft.

Programozási módszerek a feldolgozási idők rövidítésére

Ha gyakran kell hosszú futási idejű programokat végrehajtani, csökkenthetjük a végrehajtási időt. Ilyenkor a billentyűzést gyorsabb és ésszerűbb programműveletekké kell alakítani.

A programfutásban az elágazás igényli a legtöbb időt. Ha tehát az elágazási utasítások számát minimumra csökkentjük, akkor a program gyorsabban fut le. Ezen az alapon, ha a tárolókapacitás megengedi, az alprogramokat folyamatos utasításokkal lehet helyettesíteni, ezáltal lényegesen megnövekszik a feldolgozási sebesség.

Egy programlépés vagy abszolút módon 3-jegyű címmel, vagy egy programcímkevel címezhető. Abszolút címzésnél a programmutató azonnal beáll az új tárolóhelyre. Ha címkét használunk, a gépnek először meg kell keresnie ezt a tárolóhelyet. Ez a címkérés a programtárolóban mindig a 000 tárolóhellyel kezdődik. A gép végigmegy a programtárolón, amíg a kívánt tárolóhelyet meg nem találja. Ettől a helytől kezdve folytatódik a végrehajtás.

Ha egy programot először írunk a gépbe, nem ismerhetjük előre az abszolút címeket. Átdolgozás során a program-címek változnak és ez a program beírást megnehezíti. Legcélszerűbb a program első beírásakor címkéket használni és csak a program kijavítása után áttérni az abszolút címek használatára. A címkék módosítása abszolút címekre egyszerűvé válik a 2nd Nop utasítások használatával.

A 2nd Nop utasítás nem végez semmiféle műveletet, ha egy programban előfordul. Minthogy ez az utasítás a végrehajtásra nincs hatással (kivéve ha címkéként alkalmazzuk), üres utasításként is használhatjuk a közbeeső helyek szabadonhagyása érdekében, ha ez a rendelkezésre álló tárolókapacitás keretein belül lehetséges.

Ezt a módszert az alábbi ábra mutatja.

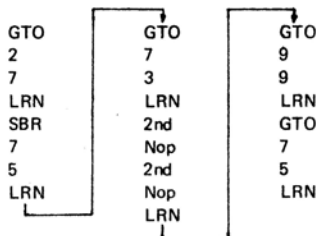
Címkézett címzés átalakítása abszolút címzésre.

	.	.	.	.
027	SBR	027	SBR	
028	Inx	028	0	
029	2nd Nop	029	7 5	
	.		.	
	.		.	
	.		.	
073	2nd Lbl	073	2nd Nop	
074	Inx	074	2nd Nop	
	.		.	
	.		.	
	.		.	
099	GTO	099	GTO	
100	Inx	100	0	
101	2nd Nop	101	7 5	

Ügyeljünk arra, hogy abszolút címként a 075 tárolóhelyet használjuk, mivel egy címkés címhez való elágazás a programmutatót a címke utáni első tárolóhelyre állítja be!

Ezenkívül gondoljuk meg, hogy a 075 abszolút cím nem 3, hanem csak 2 tárolóhelyet foglal el! Az első számjegyet a 0-t a szokásos billentyűkódban az elágazási utasítás utáni első helyen tároljuk. A második és a harmadik számjegyet kétértékű kóddá (75) foglaljuk össze és ez kerül a következő programtárolóhelyre.

Magát az elágazási utasítást szintén újra be kell írni, hogy a készüléket utasítsuk a címek automatikus összevonására. Alkalmazzuk a következő utasítássort, hogy az előző példában szereplő programot abszolút címzésre állítsuk át.



V. A KÉSZÜLÉK TULAJDONSÁGAINAK ÉS FELADATAINAK RÉSZLETES ELEMZÉSE

A következőkben részletes elemzést adunk a készülék sajátosságairól. Ebben a fejezetben megtalálhatók a részletek, ha az alapok már ismertek. Aki a készüléket még nem ismeri eléggé, annak ajánlatos ezen kézikönyv előző fejezeit ismételtelen átnézni.

A műveletekkel és függvényekkel kapcsolatos eddigi ismertetés mind a billentyűzettel való számításokra, mind ezen műveleteknek programokban való alkalmazására vonatkozik.

ALAPMŰVELETEK

Alapkijelzés

A kijelző a készülék bekapcsolt állapotának jelzésén kívül teljes számjegyes információt ad; szükség esetén negatív előjellel, tizedesponnttal, továbbá villog „túlcsordulásnál” és „lecsordulásnál”, valamint hibafeltétel esetén. Egy beírás legfeljebb 10 jegyű lehet. Az ezt követő számjegyekről a készülék már nem vesz tudomást.



Negatív szám kijelzésekor a negatív előjel közvetlenül a szám előtt baloldalt áll.

- 2.8

A kijelzett értékek pontosságával kapcsolatos információkat a C függelék tartalmazza.

Adatbeíró billentyűk

Ezeket a billentyűket a hatékonyság érdekében különválasztottuk a többtől. A műveletek többsége könnyen felismerhető, azonban vannak kivételek. A következő tájékoztatás és néhány példa segít a készülék kezelésének elsajátításában.

Számjegybilleentyűk (0 ... 9): a számjegyek beírása 0 ... 9-ig.

Tizedesponnt (.) : a tizedesponnt beírása. Szükség szerint kell beírni. Ha nem írjuk be, a számtól jobbra automatikusan megjelenik, míg egy műveleti vagy függvény billentyűt meg nem nyomunk. 1-nél kisebb számok esetén a tizedesponnt előtt 0 jelenik meg, ha csak nem használjuk a tíz rendelkezésre álló helyet. A tizedesrészek utáni 0-t a készülék általában nem jelzi ki. A gép csak az első nulla beírását veszi figyelembe, a továbbiakat nem. Ha a tizedesvesszőt közvetlenül egy kitevő után írjuk be, a mantissa és annak előjele ismét változtatható.

A π szám (2nd π). A π beírása 13 számjegyes pontossággal (3,141592653596) valósul meg a számítások elvégzésére, de a kijelzőben természetesen csak a 10 jegyre kerekített érték jelenik meg. A CE billentyűvel a π értékét nem töröljük, de helyette más szám írható be.

Előjelváltás (+/-). Ez az utasítás a kijelölt szám előjelét változtatja meg. Ha a +/- billentyűt az EE vagy a kitevő beírása után nyomjuk be, a kitevő előjele megváltozik.

Pozitív szám beírása egyszerűen a megfelelő billentyűkkel balról jobbra olyan sorrendben történik, ahogy a számot leírjuk. Egy-egy szám beírásakor a kijelzőben levő számok egy hellyel balra tolódnak. Egyetlen számjegy beírásakor a készülék csak az első bebillentyűzött tizedesponthoz fogadja.

PI, $7,892 - \pi + (-2) = 2,750407346$

Billentyű	Kijelzés
7,892 -	7,892
2nd π	3,141592654
+	4,750407346
2 +/-	-2
=	2.750407346

Törölő műveletek

A beírás törlése (CE = clear entry). Töröl minden beírást, amelyet számjegy-, tizedespont- vagy előjelbillentyűvel végeztünk; de csak akkor, ha előtte nem nyomtunk meg semmilyen függvénybillentyűt. A számítási eredményeket a tárolóból előhívott számokat, vagy a π számot a CE nem törli. Szükség esetén a billentyű leállítja a kijelző villogását. A befejezetlen műveletekre a CE utasításnak nincs hatása.

Törlés (CLR = clear). Törli a pillanatnyi számításokat és a kijelzést. Ez az utasítás megszünteti a hatványkitevős alakot, és azt ismét alapkijelzésre váltja. A kijelzés villogását is leállítja. A törlőbillentyűnek nincs hatása az adat- és programtárolók tartalmára, a t-tárolóra, a szögátszámító üzemmódra, a műszaki kijelzésre, a rögzített pontos kijelzésre és a tárolómezők elosztására.

A legtöbb számítás után a gép automatikusan törli magát. Ha egy számítás végén az = billentyűt megnyomjuk, megjelenik az eredmény a kijelzőben és a készülék készen áll újabb számításokra, a törlőbillentyű alkalmazása nélkül. Az adattároló tartalma nem törlődik automatikusan.

Programtörlés (2nd CP = clear program). Törli az összes programtárolóhelyet, a védett adatokat is beleértve; tehát az alprogramvisszaléptető tárolót és a t-tárolót, visszalépteti az összes jelzött és a programtárolót 000-ra állítja, ha az utasítást kézzel billentyűztük be. Ha a törlő utasítást egy programba építjük be, csak a t-tárolót törli.

Az adattároló törlése (2nd CMs = clear memories) olyan utasítás, amely az összes pillanatnyilag betöltött adattárolót törli.

Kettős rendeltetésű billentyűk (2nd és INV)

Készülékünk legtöbb billentyűje két feladatot lát el. Az első rendeltetés jele közvetlenül a billentyűn, a másodiké a billentyű fölött látható. A billentyűre írt feladat végrehajtásához egyszerűen meg kell nyomni a szóbanforgó billentyűt. A második feladatkör végrehajtásához a jel alatti billentyűt kell a 2nd billentyű után megnyomni. Pl. egy szám természetes logaritmusát az $\ln x$ billentyű megnyomásával számoljuk. Ezzel szemben a tízes alapú logaritmust a 2nd $\ln x$ ($\rightarrow 2nd \log$) billentyűzéssel számoljuk. A második rendeltetés megkülönböztetésére a könyvben a 2nd log jelölést használtuk. Ha a 2nd-t egymás után kétszer nyomjuk meg, vagy a 2nd utáni billentyűnek nincs másik rendeltetése, a számológép automatikusan az alaprendeltetéssel számol.

2nd-hez hasonlóan az invertáló billentyű (INV) is a számolókapacitást növeli a billentyűk számának növelése nélkül. Ha az INV billentyűt egy másik billentyű elé iktatjuk, megfordítja annak értelmét. A szükséges utasítás végrehajtására az invertáló billentyűt a következő billentyűkkel együtt alkalmazzuk:

utasítás	inverz utasítás	utasítás	inverz utasítás
EE	az EE feloldása	D.MS	tizedes fokérték
Eng	az Eng kijelzés feloldása	$P \rightarrow R$	$R \rightarrow P$
Fix	a Fix feloldása	$\Sigma +$	$\Sigma -$
log	10^x	$\bar{x}$	szórás
$\ln x$	e^x	list	adattároló felsorolás
y^x	$x \sqrt[y]{}$	SBR	visszaléptetés
Int	törfész (az egészek törlése)	$x = t$	$x \neq t$

utasítás	inverz utasítás	utasítás	inverz utasítás
sin	arc sin	$x \geq t$	$x < t$
cos	arc cos	If flg	If no „flag”
tan	arc tg	St flg	„reset flag”
Prod	osztás	Dsz	átugrás nem 0-nál
SUM	kivonás	Write	olvasás (read)

Az INV billentyű alkalmazását a programokban a megfelelő programfejezeteknél részletezzük. Az INV utasítást a billentyű kétszeri megnyomásával szüntethetjük meg, ha közben más billentyűt nem működtetünk. Inverzzel nem rendelkező utasítások esetén (pl. $|x|$, LRN) az előttünk beadott INV utasítást a gép nem veszi figyelembe. A 2nd-vel együtt működtetett utasítások invertálása a 2nd előtt és után is történhet, pl. INV 2nd log egyenértékű a 2nd INV log billentyűzéssel. Az INV alkalmazása egy meghatározott billentyűvel mindig az ezen billentyűvel foglalkozó fejezetben található.

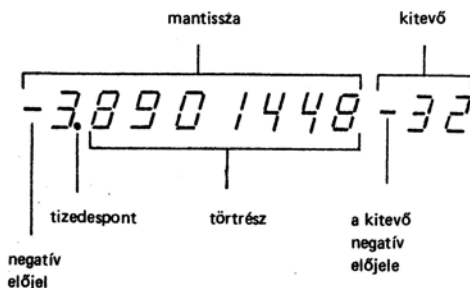
Kijelzési módok

A sokoldalú 10 számjegyű alapkijelzési mód mellett még van néhány kijelzési mód, mely a készülék műveleti tartományát és sokoldalúságát fokozza. Bár a beírás és a kijelzés legfeljebb 10 számjegyvel történhet, a belső kijelzőtároló az eredményeket mindig 13 számjegyvel tárolja. Az eredményeket a készülék csak a kijelzés érdekében kerekíti. Ezek a kiegészítő helyek megszabják a kijelzett érték tűrését, nem a pontosság növelését célozzák. Vigyázzunk ezeknek az ún. biztonsági helyértékeknek az alkalmazásánál! A pontossággal kapcsolatos felvilágosításokat a C függelék tartalmazza.

Hatványkitevős kijelzési mód

A kitevő-beírás (EE = enter exponent) olyan utasítás, amely után következő számok a 10 hatványkitevői. Az EE billentyű megnyomása után a készülék a további eredményt hatványkitevős módon jelzi ki, amíg CLR billentyűt meg nem nyomjuk, vagy a készüléket ki nem kapcsoljuk. Ez a kijelzési mód megszüntethető az INV EE vagy az INV 2nd Eng billentyűzéssel is; de csak akkor, ha a kijelzett szám a $\pm 5 \cdot 10^{-11} \dots 1 \cdot 10^{10}$ tartományban van. Ha az EE billentyűt egy részeredmény vagy végeredmény után nyomjuk meg, a 11., 12., 13. helyen levő érték törlődik, a gép csak a kijelzőben maradó értéket veszi alapul a további számításoknál.

Minden szám beírható, mint egy számérték (mantissza) és a 10 megfelelő hatványának szorzata. Írjuk be a mantisszát, max. 8 számjegyvel, nyomjuk meg az EE billentyűt, majd írjuk be a kétszámjegyű kitevőt!



A számítási tartomány abszolút értéke $1 \cdot 10^{-99} \dots 9,999\,999\,9 \cdot 10^{99}$. A 0,0000000001-nél kisebb és a 9999999999-nél nagyobb számokat hatványkitevős alakban kell beírni. Beírás során először bebillentyűzzük a legfeljebb 8 jegyű mantisszát, előjellel együtt, megnyomjuk az EE billentyűt, majd beírjuk a 10 kitevőjét annak előjellel együtt. Ha a számítási eredmények az előzőleg megadott határokat túllépik, a gép automatikusan átkapcsol hatványkitevős kijelzésre.

Példa: a 320 000 000 000 szám írható mint $3,2 \cdot 10^{11}$, a következő módon.

Billentyű	Kijelzés
CLR	0
3,2	3,2
EE	3,2 00
11	3,2 11

Az EE billentyű megnyomása után kettőnél több számjegy is beírható, de kitevőként csak az utolsó két szám érvényes. Ezzel kezünkben van a hibásan beírt kitevő javításának lehetősége, anélkül, hogy a teljes beírást törölnünk kellene a CE billentyűvel. A pozitív kitevő azt mutatja, hány helyet kell jobbra tolni a mantissa tizedespontját. Negatív kitevő a balra való eltolást jelenti. Függetlenül attól, hogy milyen formában adjuk be a hatványkitevős alak mantissáját, a készülék a számot olyan alakra hozza, hogy a tizedesponttól balra csak egy egész értéket hagy meg, ha megnyomunk egy függvény- vagy műveleti billentyűt. Pl. a $6025 \cdot 10^{20}$ beírásakor:

Billentyű	Kijelzés
CLR	0
6025	6025
EE	6025 00
20	6025 20
+	6,025 23

Hatványkitevős alak esetén a mantissa legfeljebb 8 jegyű lehet, hogy a kijelzőben hely maradjon a kitevő számára. Egy számításból adódó mantissa kijelzése is csak 8 számjeggyel történik, de a készülék 13 számjegyet tárol. Ezt a 13 jegyű értéket használja a gép a további számításokhoz. A pontosságról részletesebben a C függelékben írunk.

Megjegyzés: ha mantissaként 8-nél több számot írtunk be, akkor a hatványkitevős alakra való azonnali átérés nem lehetséges, még ha az EE billentyűt meg is nyomjuk. Ha az EE billentyűt úgy nyomjuk meg, hogy a kijelzőben 8-nál több szám van, a hatványkitevős alakra való áttérés csak egy műveleti vagy függvénybillentyű megnyomása után történik meg.

Az előjelváltó billentyűt akkor használjuk, ha a mantissának vagy a kitevőnek negatív előjelet akarunk adni. A mantissa vagy a kitevő beírása után nyomjuk meg egyszerűen a +/- billentyűt, hogy a meglevő előjelet átváltsuk! Ha a mantissa előjelét akarjuk átváltani, vagy annak törtrészéhez további számokat akarunk írni az EE billentyű megnyomása után, javítsunk a tizedespont billentyűvel, majd írjuk be a mantissa előjelét és a törtrész utólagos számjegyait!

Példa: írjuk be a $-4,962 \cdot 10^{-12}$ számot, javítsuk a kitevőt, és egészítsük ki a mantissa tizedesrészét $-4,96236 \cdot 10^{-12}$ -re!

Billentyű	Kijelzés	Megjegyzés
CLR	0	
4,962 +/-	-4,962	a mantissa és az előjel beírása
EE	-4,962 00	
12 +/-	-4,962-12	a kitevő és előjelének beírása
+/-	-4,962 12	a kitevő előjelének megváltoztatása
• ; +/-	4,962 12	a mantissa előjelváltása
36 +/-	-4,96236 12	a mantissa kiegészítése

A hatványkitevős és az alakijelzéses számok keverhetők. A számítások helyes végrehajtása érdekében a készülék a beírt adatokat átalakítja. Ha az EE billentyűt megnyomjuk, a készülék az összes eredményt hatványkitevős alakban adja meg; egészen addig, amíg a CLR, INV EE vagy INV 2nd Eng billentyűzést nem hajtunk végre, vagy amíg a készüléket ki nem kapcsoljuk. A CE törli a hatványkitevős alakot, de magát a kijelzést nem változtatja, csak a beírást.

$$Pl, 1,816 \cdot 10^3 - 581,432191 = 1,2345678 \cdot 10^3 = 1234,567809.$$

CLR	0
1.816 EE	1.816 00
3 -	1.816 03
581.432191 =	1.2345678 03
INV EE	1234.567809

A hatványkitevős alak feloldására nyomjuk meg az INV EE billentyűt, és ha a szám nincs a $+1 \cdot 10^{10} \dots +5 \cdot 10^{11}$ tartományban, a készülék ismét átkapcsol alapkijelzésre, ha a számítási eredmény is a kijelzhető tartományba esik.

$$Pl, (7 \cdot 10^{11} + 5 \cdot 10^{10}) : 25 : 25 = 1200000000$$

Billentyű	Kijelzés
7 EE	7 00
11 +	7. 11
5 EE	5 00
10 = INV EE	7.5 11
÷	7.5 11
25 = ÷	3. 10
25 =	1200000000.

A készülék automatikusan csak akkor kapcsol át hatványkitevős kifejezésre, ha az abszolút érték 9 999 999 999 feletti vagy 0,000 000 000 1 alatti. Ezért 0,1-nél kisebb abszolút érték esetén – a pontosabb kijelzés érdekében – célszerű az EE billentyű megnyomásával áttérni a hatványkitevős kijelzésre.

Ha bevezetjük a hatványkitevős alakot anélkül, hogy a számolás során megnyomnánk az EE billentyűt, az alapkijelzésre való visszaállítás automatikusan megtörténik, amennyiben az eredmény értéke lehetővé teszi. A számítási eredmények hatványkitevős alakra történő átalakítására két lehetőség van.

1. Megnyomjuk a X 1 EE = billentyűt, ezzel a kijelző tárolójában levő számot $1 \cdot 10^0 = 1$ -gyel szorozzuk, és a kijelzőt hatványkitevős alakra kapcsoljuk át. A teljes 13-jegyű szám megmarad.
2. Nyomjuk meg az EE = billentyűt. Vigyázzunk, mert ezzel a módszerrel egyben utasítjuk a gépet, hogy a kijelzőben levő kerekített értéket használja a további számításokhoz!

Mellőzzük az = billentyűvel együttes kijelzési utasításokat a számítás közben, mert az = utasítás az összes befejezetlen számítást lezárja. Ezek elkerülésére a fenti átalakítási módszereket csak a számítások után szabad alkalmazni; vagy pedig szorozzuk az X 1 EE utasítással, amelyet más műveleti utasítás után is használunk!

Műszaki kijelzési mód

A 2nd Eng utasítás segítségével hatványkitevős kijelzésnek egy módosított alakját kapjuk. Ebben a kijelzési formában a szám a mantisszából és kitevőből áll, és az elrendezés olyan, hogy a kitevő a 3 többszöröse (10^{12} , 10^{-6} stb.) és a mantissa 1, 2 vagy 3 egész számjegyű.

Pl. számítsuk ki egy huzal átmérőjét μm -ben ($1 \mu\text{m} = 10^{-6} \text{ m}$), ha a kerülete $3 \times 10^{-3} \text{ m}$.

$k = \pi d$	$d = \frac{k}{\pi}$
Billentyű	Kijelzés
CLR 2nd Eng	0. 00
3 EE	3 00
3 +/- ÷	3.-03
2nd π =	954.92966-06

Az INV 2nd Eng billentyűzéssel ez a kijelzési forma visszaállítható. Törlési utasítás vagy az INV EE billentyűzés erre azonban nem alkalmas.

Rögzített pontos beállítás.

Az alap-, a hatványkitevős és a műszaki kijelzési módban megválaszthatjuk a tizedespont utáni számjegyek

számát. Nyomjuk meg a 2nd Fix-billentyűt és utána a kívánt tizedes számjegy számát (0...8). Ezzel arra utasítjuk a készüléket, hogy az összes kijelzett eredményt a választott számú tizedes értékre kerekítse. Ez a kerekítés csak a kijelzőre érvényes, a kijelző tárolójára nem. Tehát a gép a további számításokban a teljes, kerekítetlen értéket használja.

A rögzített pontos választást meg lehet szüntetni az INV 2nd Fix utasítással. Az adatbeírás a korábbiak szerint 10 jegyű lehet, hatványkitevős esetén 8 jegyű, és az összes további számítás a 13 jegyű kerekítetlen értékkel folytatódik. Ez alól kivétel a szögek átszámítása, amikor is a kijelzőben levő értékkel számol a gép. Csak a kijelzőben levő tizedesjegyek száma változik a kívánt módon, ha az EE INV EE billentyűzést végezzük.

Ügyeljünk arra, hogy az előző példában használt műszaki kijelzést feloldjuk!

Pl. $2/3 = 0,666666667$

Billentyű	Kijelzés
2 ÷	2.
3 =	.666666667
2nd Fix 5	0.66667
2nd Fix 2	0.67
2nd Fix 0	1.
INV 2nd Fix	.666666667

Gondoljuk meg, hogy a kijelző tárolójában levő értéket a gép a kívánt alakra kerekíti!

Pl. $1 \cdot 10^{-3} : 2 = 0,0005$

Billentyű	Kijelzés
1 EE	1 00
3 +/- ÷	1.-03
2 =	5.-04
2nd Fix 2	5.00-04
INV EE	0.00*
2nd Fix 3	0.001
2nd Fix 4	0.0005
2nd Fix 5	0.00050

* Figyeljünk meg, hogy az a nulla, amelyik kb. a példa közepén a kijelzőregiszterben előfordul, nem valódi nulla. Az értékerekítés 2 tizedesjegyre nemcsak a kijelzett alakra vonatkozott! Gondoljunk mindig arra, hogy a tizedesjegyek kiválasztásával a kijelző-tárolót nem befolyásoljuk!

Villogó kijelzés

A kijelző akkor villog, ha a számológép számítási határait túlléptük, vagy ha a gépet egy nem megengedett műveletre utasítjuk. A CE billentyű lenyomásával a pillanatnyi számítások zavarása nélkül állíthatjuk le a villogást. Ezután folytathatjuk a számítást, ha a kijelzett szám még használható. A villogás a CLR billentyűvel is leállítható, de ezzel a pillanatnyi számításokat és a kijelzést is töröljük. A hibajelzés, továbbá a túlcsordulás és lecsordulás feltételeinek és követelményeinek teljes felsorolását a C. függelékben találhatjuk.

SZÁMTANI MŰVELETEK

A számok és műveletek beírási módszerei a legtöbb esetben megfelelnek a matematikai írásmódnak. A gép emlékszik a műveletekre, és szükség esetén tárolja is azokat az algebra szokásos szabályai szerint történő végrehajtásig.

Az alapműveletek +, -, X, ÷, =

Az egyszerű összeadás, kivonás, szorzás és osztás a gépnek olyan sorrendben billentyűzhető be, mintha leírnánk azokat.

$$\text{Pl. } 1,6 \cdot 10^{-19} \cdot 6,025 \cdot 10^{23} = 9,64 \cdot 10^4$$

Billentyű	Kijelzés
CLR	0
1.6 EE	1.6 00
19 +/- X	1.6-19
6.025 EE	6.025 00
23 =	964 04

Ügyeljünk arra, hogy az = billentyűvel a számtani műveleteket lezárjuk, és utasítást adunk a végeredmény kijelzésére! Új utasítással kezdetén a CLR billentyűvel a pillanatnyi számításokat töröljük.

Ügyelni kell arra, hogy ne maradjanak vissza befejezetlen műveletek a korábbi számításokból! A törlési utasítás felesleges, ha az előző feladatot = billentyűvel zártuk le. Egy számbelírás után az = művelet hatása ugyanolyan, mintha CLR-t adnánk be, azzal a különbséggel, hogy az = sem a hatványkitevős alakot, sem a kijelző villogását nem szünteti meg.

Ha a +, -, X, ÷, y^x műveleti billentyűk közül bármelyik kettőt közvetlenül egymás után nyomjuk meg, a kijelző villog. Úgysszintén, ha a fenti műveleti billentyűk egyikét = vagy jobboldali zárójel () követi, vagy a műveleti billentyűk előtt baloldali (()) zárójelet írunk. A hibafeltételek részleteit a B. függelékben találjuk.

Egy előző számításból adódó végeredményt felhasználhatunk úgy is, mint a következő számítás első adatát.

Szükségtelen a számot újra bebillentyűzni

$$\text{Pl. } 1,84 + 0,39 = 2,23 \text{ és utána } (1,84 + 0,39) : 365 = 0.006109589$$

Billentyű	Kijelzés	Megjegyzés
1.84 +	1.84	
.39 =	2.23	1.84 + 0.39
÷	2.23	
365 =	0.006109589	2.23 ÷ 365

Az algebrai műveleti rendszer

Az algebrai műveleti rendszer lényeges ismertetőjegye az algebrai műveletek sorrendje. A gépbe beprogramozták az algebrai sorrend ismert szabályait, hogy a műveleteket a gép heiesen értelmezze. Az algebrai szabályokból következik egy műveleti sorrend. Műveleti sorrend nélkül az olyan kifejezések mint pl. $5 \cdot 4 + 3 \cdot 2$ nem lennének egyértelműek. A műveleti sorrendtől függően $5 \cdot (4 + 3) \cdot 2 = 70$ vagy $(5 \cdot 4) + (3 \cdot 2) = 26$ stb. adódik. A szabályok értelmében a szorzást mindig az összeadás előtt kell végrehajtani. Az algebrailag helyes választ tehát $(5 \cdot 4) + (3 \cdot 2) = 26$.

A matematikai műveletek rendje:

1. matematikai függvények,
2. hatványok (y^x), gyökök ($\sqrt[y]{x}$),
3. szorzás, osztás,
4. összeadás, kivonás,
5. egyenlőségi utasítás.
1. Matematikai műveleteknél ill. függvényeknél a kijelzett értéket a billentyűzés után azonnal követi a függvényérték (trigonometrikus, logaritmus-, négyzet-, négyzetgyök-, exponenciális függvény), egész és tört-számokkal végzett műveletek, abszolútérték, reciprokérték számítása
2. A hatványok (y^x), gyökök ($\sqrt[y]{x}$) számítása.
3. A szorzás és az osztás.
4. Az összeadást és a kivonást csak az összes többi művelet lezárása után végezzük el.
5. Az egyenlőségi utasítás zárja le a befejezetlen műveleteket a megadott sorrendben.

Valamely művelet egy másikat egyenrangú vagy alacsonyabb sorrendű fokkal zár le. A készülék ismeri ezt a szabályt és minden egyes bebillentyűzött feladatnál alkalmazza. Egyes műveleteket azonnal végrehajt, másokat befejezetlenül hagy, amíg azokat a szabályok szerint le nem zárhatja.

Figyeljünk a sorrendet a következő példán!

$$4 : 5^2 \cdot 7 + 3 \cdot 0,5^{\cos 60^\circ} = 3,241320344$$

Billentyű	Kijelzés	Megjegyzés
4 ÷	4.	(4:) tárolva
5 x ²	25.	(5 ²) külön függvény lesz; (x ²) kiszámítása azonnal történik
X	0.16	(4:5 ²)·x kiszámítása x-szel azonos elsőbbségű
7 +	1.12	az x elsőbbsége magasabb a +-nél, tehát (4:5 ² ·7) kiszámítása és a 1,12+ tárolása
3 X	3.	(3X) tárolása
.5 y ^x	0.5	a 0,5y ^x tárolása
60 2nd cos	0.5	a cos 60° kiszámítása azonnal
=	3.241320344	a folyó műveletek lezárása; először a 0,5 cos 60° értékét számítja, ezt követi a 3·0,5 ^(cos 60°)

A kifejezés beírásakor a gép helyesen értelmezi a $[(4:5^2) \cdot 7] + [3 \cdot 0,5^{(\cos 60^\circ)}]$ kifejezést.

Lényeges szempont, hogy a műveletek szigorúan a viszonylagos elsőbbség szerint folynak le a szabályok szerint. A gép megőrzi az összes tárolt műveletet, és a végrehajtáskor az egyes műveleteknél a hozzájuk rendelt számot hívja be, mindig a kellő időben és helyen. Ha már megismerkedtünk a műveletek sorrendjével, a legtöbb feladat nehézségek nélkül megoldható, nem utolsósorban a közvetlen beírási módszer miatt. Zárójelek alkalmazásával újabb lehetőségünk van arra, hogy a feldolgozási sorrendet ellenőrizzük.

Zárójelek

Vannak olyan műveletsorok, melyeknél pontos utasítást kell adni a gépnek, hogyan számítsa ki a helyes eredményt. A számokat és a műveleteket zárójelekkel csoportosíthatjuk. A számokat és műveleteket zárójelben írva arra utasítjuk a gépet, hogy ezt a kifejezést dolgozza fel először, vagyis egyszerűsítse azt egyetlen számmá és csak ezután folytassa.

A zárójelek előnyét a következő kísérlettel világítjuk meg: nyomjuk meg a (5 X 7) billentyűket és a 35 érték jelenik meg a kijelzőben. A készülék kiszámította az 5·7 értéket és ezt a kifejezést helyettesítette 35-tel, bár az = billentyűt nem nyomtuk meg. A zárójelek ilyen rendeltetése miatt az algebrai szabályok a műveleti sorrendben együtt érvényesek minden egyes zárójelpárra. A készülék megőrzi minden műveletet, és kiszámít minden részkifejezést, ha minden szükséges információ rendelkezésre áll. Ha jobboldali zárójelet írunk be, a gép a megfelelő nyitott zárójelig visszamenőleg zárja le a műveletsort. Célzerű akkor is zárójeleket használni, ha kétségeink vannak afelől, hogyan fogja a gép a kifejezést feldolgozni.

Bár az összegek szorzását (3+2) (4+5) alakban írjuk, itt be kell billentyűznünk a szorzási utasítást, hogy a művelet végrehajtható, ugyanis a készülék nem végez el olyan műveleteket, amelyeket így módon jelölünk.

Például írjuk be a $4 \cdot (5+9) : (4-4)^{(2 \cdot 3)} = 0.2304526749$ kifejezést, és kövessük a számítás menetét!

Billentyű	Kijelzés	Megjegyzés
4 X (	4.	(4·) tárolva, a zárójeles kifejezés számítása befejezetlen marad
5 +	5.	(5+) tárolva,
9)	14.	az (5+9) számítása,
÷	56.	a műveleti sorrend szerint a (4X14) számítása,
(	56.	(56:) tárolva, a zárójeles kifejezés számítása befejezetlen marad
7 -	7.	a (7-) tárolva,
4)	3.	a (7-4) kiszámítása,
y ^x (	3.	a kitevő előkészítése
2 +	2.	
3)	5.	a (2+3) kiszámítása
=	.2304526749	a (7-4) ^(2·3) kiszámítása, majd annak eredményével osztjuk a 4·(5+9) kifejezést.

A tárolható műveletek és a hozzájuk tartozó számadatok száma korlátozott. A valóságban egyidejűleg legfeljebb 9 zárójelet szabad nyitva tartani, és legfeljebb 8 művelet lehet egyidejűleg befejezetlen, de csak rendkívül összetett feladatoknál közelítjük meg ezt a határt. Ha 9-nél több zárójelet nyitunk, vagy ha a gépnek 8-nál több műveletet kellene tárolni, a kijelző villogni kezd.

Pl. $5 + \{8 : [9 - (2 : 3)]\} = 5,96$.

Billentyű	Kijelzés	Megjegyzés
5 + (	5.	
8 ÷ (	8.	
9 - (	9.	
2 ÷ 3)	.666666667	(2 : 3) kiszámítása
)	8.333333333	[9 - (2 : 3)] kiszámítása
)	0.96	8 : [9 - (2 : 3)]
=	5.96	5 + {8 : [9 - (2 : 3)]}

Mivel az = billentyűnek az a tulajdonsága, hogy a befejezetlen műveleteket lezárja, ezért itt a) -billentyű háromszori használata helyett azt alkalmazzuk. Számítsuk ki ezt a feladatot ismét, de az első jobboldali zárójelet helyett nyomjuk meg = billentyűt!

Pl. $3 \cdot 4^{[2 - (\sqrt{7})]} = 4.700043401$

Billentyű	Kijelzés	Megjegyzés
CLR (	0.	
3 X (	3.	
4 y ^x (	4.	
2 y ^x (	2.	
7 INV y ^x	7.	
4)	1.626576562	$\sqrt[4]{7}$
+/-	-1.626576562	$-(\sqrt[4]{7})$
)	.3238557891	$2^{-(\sqrt[4]{7})}$
)	1.566681134	$4^{0.323}$
)	4.700043401	$3 \cdot 4^{0.323}$

Minden lezárt zárójelnél annak tartalmát a gép a következő kezdő zárójelig kiszámítja, és egyetlen értékkel helyettesíti. Ennek ismeretében meghatározhatjuk egy tetszőleges célú számítás feldolgozási menetét. Ezen belül a közbenső eredményeket is követhetjük.

„Vakműveletek” zárójelekkel

Egy további technikai fogás a zárójelekkel kapcsolatban a CE „vakművelet”. A CE billentyűvel másodszor is beírhatjuk a kijelzett értéket anélkül, hogy ezt újra be kellene billentyűzni. Különösen akkor célszerű egy értéket zárójelek közé ilyen módon bevenni, ha arra egy kifejezésben kétszer egymás után szükség van.

Kövessük a számítást a következő példában! Pl. $3,296214 + (3,296214 \cdot 6) = 23.073498$

Billentyű	Kijelzés	Megjegyzés
CLR 3.296214 +	3.296214	
(CE X	3.296214	CE újra beírja a 3.296214 értéket,
6)	19.777284	
=	23.073498	

a sokjegyű 3,296214 számot így csak egyszer kellett beírni.

ALGEBRAI FÜGGVÉNYEK

Az egyváltozós függvények könnyen érthetők és szemléletesek. Ezek a függvények a kijelző tárolójában levő értékkel dolgoznak, és ezt azonnal a függvényértékkel helyettesítik. Ezek a függvények nem befolyásolják a pillanatnyi számításokat, ezért mindenkor alkalmazhatók egy számításban belül is. Ezen függvények pontosságát egyrészt a C függelékben, másrészt a szöveg megfelelő helyén tárgyaljuk.

Reciprokérték 1/x

Ezt úgy számítjuk, hogy az 1-et elosztjuk az x-szel, ha $x \neq 0$, a kijelző villog. Pl.

Billentyű	Kijelzés
3.2 1/x	0.3125

Ügyeljünk arra, hogy a kijelzett értéket azonnal a megfelelő függvényérték helyettesíti, mielőtt egy matematikai függvény billentyűjét megnyomjuk!

Logaritmusok

Természetes alapú logaritmus ($\ln x$): valamely (a kijelző tárolójában levő) x szám természetes (e alapú) logaritmusának számítása. A kijelzés villog, ha $x \leq 0$.

10-es alapú logaritmus (2nd log): a kijelzőtárolóban levő x szám 10-es alapú logaritmusának számítása. Kijelző villog, ha $x \leq 0$.

Pl. $\log(1 + \ln 1,7) = 0.1848697249$.

Billentyű	Kijelzés
CLR	0
(1 +	1.
1.7 $\ln x$)	1.530628251
2nd log	.1848697249

A 10 és az e hatványai

A természetes numerus logaritmi ($\ln x$): az x , vagyis a kijelzőregiszterben levő érték numerus logaritmiájának számítása. A $-227,9559242 \leq x \leq 230,258509$ tartományon kívül a kijelző villog.

10-es alapú numerus logaritmi ($\ln x$ 2nd lg): a kijelzőregiszterben levő x szám 10-es alapú numerus logaritmiájának számítása. Az értéktartomány $-99 \leq x \leq 99,999999998$. Az ezen kívül eső értékeknél a kijelző villog. Pl.: $e^{(3+10^{9.9})} = 147,7116873$

Billentyű	Kijelzés
CLR (3 +	3.
.3 $\ln x$ 2nd log)	4.995262315
$\ln x$	147.7116873

Szögszámítások

Szögszámítási üzemmódok

A szögeket mérhetjük (rég) fokban, radiánban vagy újfokban (gon, grade). A derékszög $= 90^\circ = \frac{\pi}{2} \text{ rad} = 100^g$.

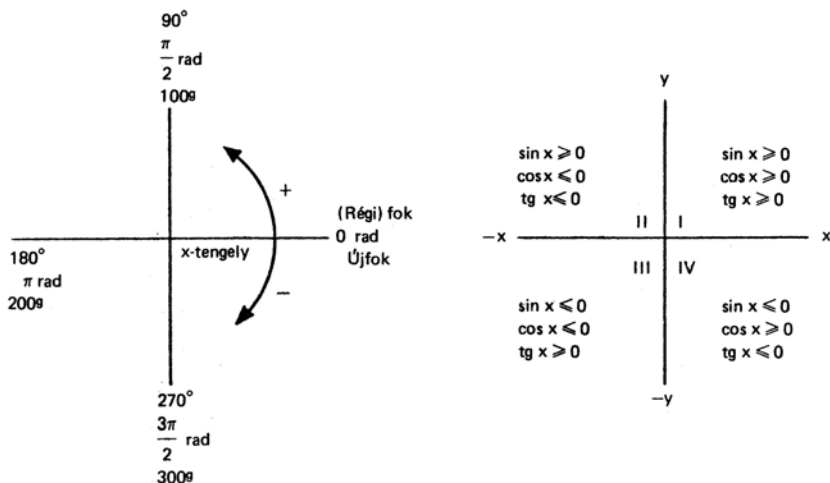
Régi fokhoz a 2nd Deg, radiánhoz a 2nd Rad, újfokhoz a 2nd Grad billentyűket nyomjuk meg. Bekapcsoláskor a készülék (rég) fok számítására áll be. Ez mindaddig megmarad, amíg másik szögegységre nem váltunk. Ha a készüléket egy meghatározott üzemmódra kapcsoltuk, akkor az összes beírt és kijelzett szögérték ebben

a mértékegységben értendő; egészen addig, amíg más mértékegységet nem választunk, vagy a készüléket ki nem kapcsoljuk. A CE és a CLR billentyűnek nincs hatása a szögegységre.

A szögszámítási üzemmód nincs befolyással a számításokra, kivéve a szögfüggvényeket és a poláris derékszögű koordináta-átszámításokat. A helyes szögegység megválasztása egyszerű, de igen könnyű elfelejteni. Ennek a lépésnek az elmulasztása gyakori hibaforrás olyan számológépeknél, ahol a szögegységet meg kell választani.

Szögfüggvények

Színusz, koszinusz, tangens (2nd sin, 2nd cos, 2nd tan): a színusz, a koszinusz és a tangens kiszámítása a kijelző tárolójában levő szögértékre vonatkoztatva. Az összes szögértéket az x-tengelytől számítjuk, a pozitív irány az óramutató járásával ellentétes, a negatív azzal megegyező; amint azt az ábra is szemlélteti.



A jobboldali ábrarész azt mutatja, hogy melyik szögfüggvény milyen előjelű az egyes síknegyedekben.

A szögek mérésénél gondoljuk meg, hogy minden szögnek van egy ellenkező előjelű megfelelője!

Pl. $-45^\circ = 315^\circ$

Ha a szöget fok–perc–másodperc alakban fejeztük ki, az átszámítható tizedestört alakra. Az átszámításról később szólunk, ugyanebben a fejezetben. Győződjünk meg arról, hogy a készüléket a foknak megfelelő üzemmódra kapcsoltuk-e! Kétséges esetben nyomjuk meg a 2nd Deg billentyűt!

Pl. $\sin 30^\circ 13' 48'' + \tan 315^\circ = -0,5$

Billentyű	Kijelzés
30.1348 2nd D.MS	30.23
2nd sin +	0.5034724109
315 2nd tan	-1.
=	- .4965275891

A szögfüggvény-értékeket 360° -nál nagyobb szögértékekre is kiszámíthatjuk. Amíg a szögfüggvényt alap- és nem hatványkitevős alakban jellezzük ki, addig a $\pm 36\ 000$ fok, a ± 200 radián és a $\pm 40\ 000$ újfok szögtartományban, a kijelzett értékek pontosak. Ezen a tartományon kívül a pontosság dekádanként egy számjeggyel csökken. Ha az x értékének abszolút értéke $3,6 \cdot 10^{14}$ foknál, $4,0 \cdot 10^{14}$ újfoknál, vagy $6,2799993 \cdot 10^{12}$ radiánnál nagyobb, a készülék már nem ismeri fel a függvények periódusos voltát.

A többi szögfüggvény is könnyen kiszámítható hasonló módon: $\cotg x \rightarrow 2nd \tan 1/x$, $\sec x \rightarrow 2nd \cos 1/x$ és $\csc x \rightarrow 2nd \sin 1/x$.

Árkuszfüggvények

Inverz művelet (INV). Ha valamely billentyű előtt megnyomjuk az INV billentyűt, akkor a gép a függvény inverzét képezi; szögfüggvények esetén az árkuszfüggvényekhez jutunk, pl. az $\arcsin x$ billentyűzése INV 2nd sin.

Az árkuszfüggvényeknél a kijelzett függvényértékhez tartozó szögértéket számítjuk ki. Az árkuszfüggvényből származó legnagyobb szögérték 180 fok, π radián vagy 200 újfok. Minthogy ezek a függvények nem egyértékűek (pl. $\arcsin 0,5 = 30^\circ, 150^\circ, 390^\circ$ stb.) a szögeket az egyes függvényekből a következők korlátozással számítjuk ki.

Árkuszfüggvény	A származtatott szög tartománya
$\arcsin x$	$0...90^\circ, 0... \pi/2 \text{ rad}, 0...100^\circ$
$\arcsin(-x)$	$-90...0^\circ, -\pi/2...0 \text{ rad}, -100...0^\circ$
$\arccos x$	$0...90^\circ, 0... \pi/2 \text{ rad}, 0...100^\circ$
$\arccos(-x)$	$90...180^\circ, \pi/2... \pi \text{ rad}, 100...200^\circ$
$\arctan x$	$0...90^\circ, 0... \pi/2 \text{ rad}, 0...100^\circ$
$\arctan(-x)$	$-90...0^\circ, -\pi/2...0 \text{ rad}, -100...0^\circ$

Az $\arcsin x$ és $\arccos x$ esetén érvényes, hogy $-1 \leq x \leq 1$; ezen a tartományon kívül a kijelző villog.

Pl.: $\frac{\pi}{4} + \arctg(0,2 \cdot \pi) = 1,34638028$.

Billentyű	Kijelzés
2nd Rad	0
2nd $\pi \div$	3.141592654
4 +	.785391634
(.2 X 2nd π)	.6283185307
INV 2nd tan	.609821161
=	1,34638028

A radián szögegység megválasztása a X 2nd tan billentyűzés előtt bárhol történhet. Ennek ellenére célszerű a szögegységet még a számítás előtt megválasztani, hogy megbizonyosodjunk a helyes beállításról; ahelyett, hogy menetközben a feladatmegoldás helyett ezzel foglalkozzunk. Teljesen mindegy, hogy mikor választunk szögegységet, mert ennek csak a szögszámításokra van hatása. A többi árkuszfüggvényt a következők szerint kell kiszámítani:

$\arccot x \rightarrow 1/x$ INV 2nd tan

$\operatorname{arcsec} x \rightarrow 1/x$ INV 2nd cos

$\operatorname{arccosec} x \rightarrow 1/x$ INV 2nd sin.

Szög—radián—újfok átszámítások

Gyakran kell a szögértékeket átszámítani egyik mértékegységről a másikra. Az átszámítási tényezőket az alábbi táblázat mutatja:

-ról \ -ra	Szög	Radián	Újfok
Szög	1	$\frac{\pi}{180}$	$\frac{1}{0,9}$
Radián	$\frac{180}{\pi}$	1	$\frac{200}{\pi}$
Újfok	0,9	$\frac{\pi}{200}$	1

Billentyű	Kijelzés	Megjegyzés
120 X 2nd π ÷	376.9911184	
180 = X	2.094395102	radián
200 ÷ 2nd π =	133.3333333	űfok
X	133.3333333	űfok
.9 =	120.	űfok

Mivel az átszámítások a készülék szövegegyységétől függetlenek, különösen vigyázni kell, ha az eredményeket további számításokban is felhasználjuk! A szövegegyiséget úgy kell megválasztani, hogy az az eredmény szövegegységével összhangban legyen!

Egészszám és abszolút érték képzése

Egészszámképzés (2nd Int). Ezzel az utasítással töröljük a kijelzett szám tört részét. Ugyanakkor az INV 2nd Int utasítással a kijelző tárolójában levő szám egész részét töröljük.

Abszolút érték képzése 2nd |x|: a kijelzőben levő szám mindig pozitív előjélű lesz. Példaként számítsuk ki a -13.5 abszolút értékét és egész részét! A fenti műveletek különösen programírásnál előnyösek.

Billentyű	Kijelzés
13 +/- ÷	-13
5 =	-2.6
2nd Int	-2
2nd x	2.

Figyeljük meg, hogy az egészszám képzésének alapja nem a kijelzett érték, hanem a kijelző tárolójában levő szám! Ha tehát a 2nd és az Int billentyűt nyomjuk meg, és a kijelzőtároló tartalma 4.999999999999 (a kijelzőben 5-re kerekítve), akkor egészszámként 4 marad a kijelzőben. Ahhoz, hogy a kijelzőben az 5 értéket tarthassuk meg, a 2nd Int előtt nyomjuk meg az EE INV EE billentyűt, hogy a ki nem jelzett helyeket eltüntessük! Hatványkitevős alak esetén az egészszám-billentyűvel csak a szám tényleges tört részét távolítjuk el, nem a látszólagos tört részét. Pl. $1,2345 \cdot 10^3$ esetén a 2nd Int billentyűzés az $1,234 \cdot 10^3$ eredményt adja. Valójában 1234.5-ből 1234 lett.

Négyzetre-emelés és négyzetgyökvonás

Négyzetre emelés (x^2): a kijelző tárolójában levő szám négyzetének számítása. Ha $|x| \geq 10^{+50}$, akkor villog a kijelző.

Négyzetgyökvonás ($\sqrt{x}$): a kijelző tárolójában levő szám négyzetgyökének számítása. Ha x negatív előjélű, a kijelző villog.

Példa: $[\sqrt{3,1452} - 7 + (3,2)^2]^{1/2} = 2,239078197$

Billentyű	Kijelzés
CLR	0.
3.1452 $\sqrt{x}$ -	1.773471173
7 +	-5.226528827
3.2 x^2	10.24
)	5.013471173
$\sqrt{x}$	2.239078197

Hatványozás és gyökvonás

Hatványozás (y^x): a kijelző tárolójában levő y számot az x -edik hatványra emeljük. A beírási sorrend y y^x ; ezután egy műveleti vagy az egyenlőségjel-billentyű következik. $y < 0$, esetén a kijelző villog.

Gyökvonás (INV y^x): a kijelző tárolójában levő y szám x -edik gyökének kiszámítása. A beírási sorrend y INV y^x , amit egy műveleti vagy az egyenlőségjel billentyű követ. Ha $y < 0$ és $x = 0$, vagy ha $y = 0$ és $x < 0$, akkor a kijelző villog.

Ezek a műveletek nincsenek közvetlen hatással a kijelző tárolójára. Egy második érték beírása is szükséges, amelyet egy műveleti utasításnak kell követni, még mielőtt a matematikai műveletet a készülék végrehajtana.

Példa: $\sqrt[3]{2,36 \cdot 0,23} = 0,9362893421$

Billentyű	Kijelzés	Megjegyzés
2.36 y^x	2.36	az y beírása az y^x -hez
.23 +/-	-0.23	az x beírása az y^x -hez
INV y^x	.8207865654	az y kiszámítása az $\sqrt[y]{y}$ -hoz
3 =	.9362893421	az x beírása az $\sqrt[x]{y}$ -hoz, majd a végeredmény.

Az y^x függvényt logaritmus-függvény segítségével számíthatjuk ki. A meghatározások alapján a különböző $x-y$ kombinációkra a következő értékek adhatók:

y	x	y^x	$\sqrt[x]{y}$
0	0	1.	"1."
0	-x	"9.9999999 99"	"9.9999999 99"
0	x	0.	0.
1	0	1.	"1."
y	0	1.	"9.9999999 99"
-1	0	"1."	"1."
-y	0	" "	" "
		1.	9.9999999 99"
-y	$\pm x$	" y $\pm x$,"	" $\sqrt[\pm x]{ y }$ "

TÁROLÁS

A címezhető adattárolók biztosítják az adatok tárolását későbbi felhasználásra. Ezeket a tárolómezőket általában adattárolóknak vagy adatregisztereknek nevezik, szemben a programtárolókkal, amelyekbe a programokat visszük be. A tárolóbillentyűk nem befolyásolják a pillanatnyi számításokat, bármikor használhatók.

Szokásos körülmények között mindegy, hogy melyik adatot melyikben tároljuk. Csak a középérték és a szórás számítások foglaltak a 01..06 tárolók a belső számításokhoz. Ha statisztikai számítások közben más adatokat is kívánunk tárolni, akkor erre a célra a 01..06 tárolókat nem használhatjuk. Ha igen sok adattárolót használunk, akkor fel kell jegyeznünk, hogy melyik értéket melyikben tároltuk.

A tárolókapacitás megválasztása — a tárolómezők elosztása

Ha a készüléket első alkalommal kapcsoljuk be, 60 tároló áll az adattárolás rendelkezésére. Az adattárolók a teljes tárolókapacitás részei, ahol mind adatok, mind programok tárolhatók. A tárolókapacitás 10-es tárolócsoportokba való felosztásának lehetősége biztosítja, hogy a programtárolók és az adattárolók számának viszonyát szükség szerint módosítsuk. Írjuk be azon tízes csoportok számát, amelyeket adattárolásra kívánunk használni, és nyomjuk be a 2nd Op 17 billentyűt, pl. 20 adattároló esetén a 2 2nd Op 17-t! Ezt a felosztást a kijelző az ábra szerint mutatja.

programtároló adattároló

Ily módon azt jelzi a gép, hogy adattároláshoz 20 (00..19), programtároláshoz 800 (000..799) hely áll rendelkezésre. A különleges kijelzési módokat, azaz a rögzített pontos, a hatványkitevős és a műszaki kijelzést a tárolótartományok elosztása előtt meg kell szüntetni. Minden egyes, nem az adattároláshoz használt tároló 8 tárolóhellyel növeli a programtárolót. A 2nd Op 16 billentyűzéssel a pillanatnyi elosztást mindenkor meg lehet

vizsgálni, és a fent ábrázolt módon ki lehet jelezni. Részletesebben a tárolókapacitás elosztásáról később, a Tárolókapacitás és elosztás c. fejezetben szólnunk.

Minthogy max. 100 adattárolót foglalhatunk le, a tárolásra vonatkozó utasítás után közvetlenül a kétértékű XX tárolócímre kell beírunk. Ha elmarad a tárolási cím beírása, a kijelzőben a pillanatnyi érték villog.

Rövid alakú címzést egyjegyű címmel csak akkor végezhetünk, ha a cím 10-nél kisebb és ezt nem számjegy beírása követi.

Az adattároló törlése

Az adattároló törlése 2nd CMs (clear memories) olyan utasítás, amely a pillanatnyi elosztás szerint az összes adattárolót törli. Ennek a billentyűnek nincs hatása a T-tárolóra, a programtárolóra, a tárolási kapacitás elosztására, a kijelzésre vagy a pillanatnyi számításokra.

Tárolás és az adatok behívása

Tárolás és az adatok behívása (STO XX). A kijelző tárolójában levő értéket az XX tárolóban tároljuk és az ebben levő korábbi értékeket töröljük.

Előhívás (RCL XX): az XX tároló tartalmának előhívása és kijelzése. Ezt az értéket továbbra is tároljuk az XX tárolóban.

Példa: tároljuk a 3,012 számot a 22. tárolóban, majd hívjuk elő!

Billentyű	Kijelzés
3.012 STO 22	3.012
CLR	0
RCL 22	3.012

A tárolási utasításokkal billentyűzést takaríthatunk meg, ha sokszámjegyű számokat tárolunk, amelyeket később többször is felhasználunk.

Pl. számítsuk ki a $3x^2 - x - 7,1$ értékét, ha $x = 2,9467281$!

Billentyű	Kijelzés
CLR 3 X	3.
2.9467281 STO 12	2.9467281
$x^2 -$	26.04961949
RCL 12	2.9467281
$- 7.1 =$	16.00289139

Az x sokjegyű értékét csak egyszer kellett beírunk. A tárolásnak és a behívásnak nincs hatása a pillanatnyi számításokra.

Az adattárolók képesek arra is, hogy közbeni eredményeket és ismétlődő számokat is megőrizzenek.

Pl. számítsuk ki a következő kifejezést!

$$\frac{\sin(3x/2) - \cos(3x/2)}{x}, \quad \text{ahol } x = 20,6821776^\circ$$

Billentyű	Kijelzés	Megjegyzés
2nd CMs () 3 X	3.	
20.6821776 STO 14	20.6821776	x tárolása a 14. tárolóban
$\div 2$) STO 17	31.0232664	$3x/2$ a 17. tárolóban
2nd sin $\sim$	.5153861069	
RCL 17	31.0232664	$3x/2$ előhívása a 17. tárolóból
2nd cos) $\div$	-.3415719789	
RCL 14	20.6821776	x előhívása a 14. tárolóból
=	-.0165152812	végeredmény.

Ha olyan tárolót akarunk lefoglalni, amely a pillanatnyi elosztás szerint nem áll rendelkezésünkre, a kijelző villog.

Közvetlen műveletek az adattárolókkal

Valamely számot bármikor tárolhatunk számítás végezte közben, annak zavarása nélkül. Ezenkívül a kijelzett értékkel és az adattárolóból vett másik értékkel összeadást, kivonást, szorzást és osztást is végezhetünk, miközben a kijelzőtároló értéke nem változik. Ha ezen műveletek után a kijelző villog, az azt jelenti, hogy ebben a tárolóban túlléptük a számológép kapacitását; feltételezve, hogy nem olyan tárolót hívtunk, amely a pillanatnyi mezőelosztás határán kívül fekszik.

SUM XX. A kijelzőtároló értékét hozzáadjuk az XX adattároló tartalmához, és az eredményt az XX adattárolóban tároljuk.

INV SUM XX. A kijelzőtárolóban levő értéket kivonjuk az XX adattároló tartalmából, és az eredményt az XX adattárolóban tároljuk.

2nd Prd XX. A kijelzőtárolóban levő értékeket megszorozzuk az XX adattároló tartalmával, és az eredményt az XX adattárolóban tároljuk.

INV 2nd Prd XX. Az adattároló tartalmát elosztjuk a kijelzőtároló értékével és az eredményt az XX adattárolóban tároljuk.

Ezzel fölöslegessé válnak a hosszadalmas előhívási műveletek és az újabb tárolásra szolgáló billentyűzések.

Példa: számítsuk ki az $x^2 + 9$ kifejezés értékét $x = -1; 2$ és 3 értéknél, majd az eredményeket adjuk össze!

Billentyű	Kijelzés	3. adattároló
1 +/- x^2 +	1.	0
9 = STO 03	10.	10
2 x^2 +	4.	10
9 = SUM 03	13.	23
3 x^2 +	9.	23
9 +	18.	23
-RCL 03	23.	23
=	41.	23

Figyeljük meg, hogy a 03. tárolóban az első számítást az STO billentyűvel tároljuk! Ezt az eljárást a most ismertetett módon célszerű végrehajtani, annak biztosítására, hogy csak a szükséges értékeket tároljuk az adattárolóban. Az STO billentyű a tároló minden előző tartalmát törli, mielőtt az új értéket tároljuk. Példa: a tanulmányaikat befejező főiskolai hallgatók százalékos aránya az első évben 76,8%, a másodikban 81,3%, a harmadikban 92,2% és a negyedikben 95,9%. Mekkora a végzős hallgatók százalékos aránya, és hány százalékát fejezi be a 3. és 4. évet?

Billentyű	Kijelzés
.768 X	0.768
.813 X	0.624384
.922 STO 11 X	0.575682048
.959 2nd Prd 11 =	0.552079084
RCL 11	0.884198

Tehát a főiskolára beiratkozó diákoknak kb. 55%-a végez. Több mint 88% azoknak az aránya, akik a 3. és 4. évet elérik és végznek.

Tároló-kijelző csere

Tároló-kijelző csere (2nd Exc XX): a XX adattároló és a kijelzőtároló tartalma helyet cserél, vagyis a kijelzett értéket tároljuk, és a tárolt értéket jelezzük ki.

A cserebillentyűnek más, billentyűzést megtagarító szerepe is van. Két mennyiséget ellenőrizhetünk anélkül, hogy bármelyiket is elvesztenénk.

Pl. számítsuk ki az $A^2 + AB + 2B^2$ kifejezést, ha $A = 0,258963$ és $B = 1,255632$!

Billentyű	Kijelzés	Megjegyzés
.258963 STO 13	0.258963	A tárolása a 13. tárolóban
$x^2 + 1.255632 \times$	1.255632	a B beírása
2nd Exc 13	0.258963	az A előhívása és a B tárolása
+ 2 X	2.	
RCL 13	1.255632	a B előhívása a 13. tárolóból
$x^2 =$	3.545447504	végeredmény

KÜLÖNLEGES VEZÉRLÉSI MŰVELETEK

Az Op billentyűvel végrehajtható műveletek sora lényegesen növeli a számológép teljesítményét. Ezek közül egyes műveletek minden üzemmódban elvégezhetők, míg mások csak különleges üzemmódban, vagy a sornymutatóval összekapcsolva. Minden vezérlési műveletet a 2nd Op nn billentyűzéssel vezetünk be, ahol nn a megfelelő, hozzárendelt kétjegyű kód (rövid címzés lehetséges). A kódok rövid leírását a következő felsorolás adja, a részletes ismertetést a következő oldalakon találhatjuk.

Kód (nn)	Művelet
00 *	a nyomtató tároló előkészítése
01 *	10 számjegy beírása a kijelzőbe (a nyomtatott sor bal külső negyedére vonatkozó 5 szám- és betűkód)
02 *	10 számjegy beírása a kijelzőbe (a nyomtatott sor bal belső negyedére vonatkozó 5 szám- és betűkód)
03 *	10 számjegy beírása a kijelzőbe (a nyomtatott sor jobb belső negyedére vonatkozó 5 szám- és betűkód)
04 *	10 számjegy beírása a kijelzőbe (a nyomtatott sor jobb külső negyedére vonatkozó 5 szám- és betűkód)
05 *	a nyomtató tároló tartalmának kinyomtatása
06 *	az Op 04 utolsó 4 írásjelének és a pillanatnyi kijelzett értéknek a kinyomtatása
07 *	rajzolás (plotting) a 0..19 oszlopban, a kijelző adatai szerint
08 *	a címkek felsorolása, amelyek pillanatnyilag felhasználásra kerülnek a programtárolóban
09	a programtároló lefoglalása, előre megadott software-programmal
10	az előjelfüggvény alkalmazása a kijelzőtároló értékére
11	a szórásnégyzet kiszámítása
12	a meredekség és a tengelymetszet kiszámítása
13	a korrelációs együttható kiszámítása
14	a kijelzőben levő x értékhez tartozó új y -érték (y') kiszámítása
15	a kijelzőben levő y értékhez tartozó új x érték (x') kiszámítása
16	a pillanatnyi tárolómező-elosztás kijelzése
17	a tárolómező-elosztás változtatása
18	ha a programban nincs hibafeltétel, beiktatjuk a 7. jelzőt
19	ha a programban van hibafeltétel, beiktatjuk a 7. jelzőt
20–29	a 0...9 adattároló valamelyikének növelése 1-gyel
30–39	a 0...9 adattároló valamelyikének csökkentése 1-gyel

* csak sornymutatóval összekapcsolva.

Nyomtatási műveletek – Op 00–08

Ezek a vezérlési műveletek csak a készülékhez csatlakoztatott KA–100 típusú sornymutatóval együtt hajthatók végre. A számítási eredmények kinyomtatása növeli a számológép sokoldalú használatosságát. A vezérlési műveletek egyben bővíti a nyomtatási lehetőségeket (betűk, számok, adatok kiírása, a programban alkalmazott címkek, ill. a megfelelő programtárolócímkek felsorolása). Ezeknek a különleges vezérlési műveleteknek az alkalmazásáról részletesebb magyarázatot a VI. Nyomtatásvezérlés c. fejezetben találunk.

Egy software-program elemzése (programátvitel) — Op 09

A 2nd Pgm mm 2nd Op 09 billentyűzése az mm software-programot átveszi a programtárolóba. Ezáltal a program elemezhető, átalakítható, és egyéni kívánalnak szerint módosítható. Ügyeljünk arra, hogy a módosítás milyen hatással van az abszolút címekre, és az összevont billentyűkódokra! Ha egy software-programot beviszünk a programtárolóba, a 000 tárolási helytől kezdve kicseréli az összes korábban tárolt utasítást. Program átvitele programtárolóból software-egységbe nem lehetséges. Ha egy program bármilyen okból terjedelmesebb a programtároló kapacitásánál, nem tudjuk átvinni a tárolóba; ilyenkor a kijelző villog. A kijelző akkor is villog, ha a pillanatnyilag beiktatott egység az mm programot nem tartalmazza.

Az előlőfüggvény — Op 10

Az Op 10 billentyűzéssel a kijelzőtárolóban levő x pillanatnyi értékére alkalmazzuk az előlőfüggvényt:

x értéke	kijelzés
$x > 0$	1.
$x = 0$	0.
$x < 0$	-1.

Statisztikai számítások — Op 11–15

A 11–15 vezérlési művelet a statisztikai számításokra szolgál, amelyeket a következő fejezet Átszámítások és statisztikai számítások című részében tárgyaljuk részletesen.

A tárolási tartomány elosztása — Op 16–17

A 2nd Op 16 utasítással megadjuk a programtároló és az adattároló közötti pillanatnyi elosztást. Leválasztó tizedespont jelzi az utolsó betölthető programtároló helyet és a legmagasabb sorszámu adattárolót. A bekapcsoláskor kijelzett elosztás 479.59. A tárolómezők elosztásának megváltoztatása céljából írjuk be a szükséges adattárolók tízes csoportjainak számát (0...10), és billentyűzzük 2nd Op 17 utasítást! A készülék az elosztást a fentiek szerint jelzi ki. Olvassuk el ezenkívül az V. rész Tárolás, továbbá a Tárolókapacitás megválasztása — a tárolómezők elosztása c. fejezetet.

Vizsgálati műveletek — Op 18–19

A 18 és a 19. művelettel a pillanatnyi program hibaállapotát ellenőrizhetjük. Ha nincs hibafeltétel, iktassuk be a programba annak részeként a 2nd Op 18 Flag 7 utasítást! A 2nd Op 19 a 7. jelzőt lépteti, ha hibafeltétel van. A 7. jelző az If flag utasításra felülvizsgálja a vizsgálati utasítást. A vizsgálat eredményétől függ a további döntés. A 7. jelző nem változik, ha a vizsgálat feltételei nem teljesülnek.

A jelzőkről részletesebb tájékoztatást az V. rész Programozás c. fejezete tartalmaz.

Az adattárolók értékének növelése és csökkentése — Op 20–29 (30–39)

Az Op utasítással a 0...9 adattároló tartalmát eggyel növelhetjük vagy csökkenthetjük. A 2nd Op 2n utasítással az n tároló tartalmát eggyel növeljük, ahol n értéke 0...9 lehet; a 2nd Op 3n utasítással az n tároló tartalmát eggyel csökkentjük, ahol n értéke 0...9 lehet. Ha a fenti billentyűzések egyikét elvégezzük, vagy azokat a program tartalmazza, az n tároló tartalmát — a benne tárolt értéktől függetlenül — 1-gyel megváltoztatjuk.

Pl. a 2nd Op 34 utasítással a 4. adattároló tartalmából egyet kivonunk.

ÁTSZÁMÍTÁSOK ÉS STATISZTIKAI SZÁMÍTÁSOK

Több, gyakran előforduló matematikai művelet programját tartalmazza a gép. Ezek a műveletek lényegesen csökkentik az ismétlődő műveletek végrehajtásához szükséges billentyűzést.

Átszámítások

A poláris és a derékszögű koordináta-rendszerek közötti átszámítás a géppel közvetlenül elvégezhető. Hasonlóan számíthatunk át fok–perc–másodperc alakú szöget tizedestört alakra és viszont.

Szögek átszámítása

A fok–perc–másodperc alakú szögek átszámítása tizedestört alakra a 2nd D.MS billentyűzéssel végezhető el. Az INV 2nd D.MS billentyűzés az átszámítást megfordítja. A percek és a másodpercek értéke tetszőleges kétjegyű szám lehet. Az átszámítás csak a kijelzett értéken alapul. A fok–perc–másodperc alak beírásakor a fokok és percek közé pontot kell írni, tehát az DD.MMSSsss alakú. Esetenként a fok–perc–másodperc alakra való áttéréskor kerekítés miatt 60 perccel vagy 60 másodperccel jelez ki. Ez azt jelenti, hogy a legközelebbi fokértékre vagy másodpercekre a legközelebbi percértékre történt a kerekítés.

Billentyű	Kijelzés	Megjegyzés
47.131272 2nd D.MS	47.2202	DD.dddd
INV 2nd D.MS	47.131272	DD.MMSSsss

Itt a DD a fokokat, a dd azok tizedestört-részét, az MM a perceket, SSsss a másodperceket és azok törtrészeit jelölik. Szögfüggvényeknél a gép csak a tizedestört alakban, és nem a perc–másodpercben megadott szögértékkel számol. Tehát a fok–perc–másodperc alakú szögértéket előzőleg át kell számítani a D.MS billentyűzéssel. Ügyeljünk arra, hogy perc–másodperc beírásakor mindig két számjegyet alkalmazzunk.

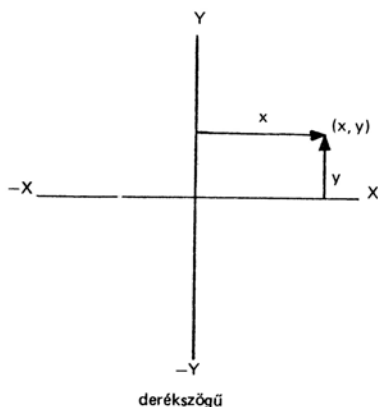
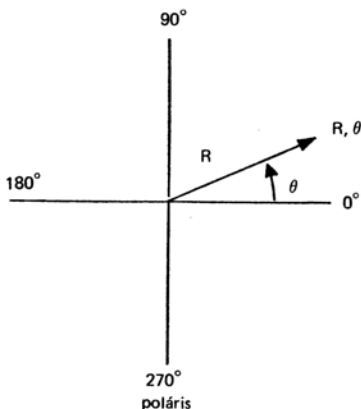
Pl. az $5^{\circ}4'3''$ csak 5,0403 alakban írandó be helyesen. A D.MS billentyű alkalmazható az óra–perc–másodperc alakról tizedestört alakra történő átszámításokhoz is.

Poláris–derékszögű koordináták átszámítása

Az $x \approx t$ billentyű a kijelző tárolójában levő x -értéket és a t -tárolóban levő t -értéket felcseréli. Ez a t -tároló független az összes többi adattárolótól, de lényegében adattárolóként működik. Csak az $x \approx t$ billentyűvel lehet hozzáférni. A t -tárolót a CP törli, ha a programban is szerepel.

A poláris–derékszögű koordináták átszámítása a 2nd P→R billentyűzéssel valósul meg. Az INV 2nd P→R utasítással a derékszögű koordinátákat poláris koordinátákra számítjuk át. A számításoknál 4 belső műveletre van szükség.

A poláris–derékszögű átszámítást akkor alkalmazzuk, ha az r távolsággal és θ irányszöggel megadott poláris koordinátákat a szintén egy pontot meghatározó x abszcissza és y ordináta révén adott derékszögű koordinátákra akarjuk átszámítani.



Billentőzűs a koordináták átszámításához:

poláris – derékszögű

az R beírása

az $x \approx t$ billentőzűsése

az θ beírása

2nd P→R billentőzűsése az y kijelzésére

$x \approx t$ billentőzűsése az x kijelzésére

derékszögű – poláris

az x beírása

az $x \approx t$ billentőzűsése

az y beírása

INV 2nd P→R billentőzűsése a θ kijelzésére

$x \approx t$ billentőzűsése az R kijelzésére.

Ellenőrizzük a θ szövegységét beírás előtt a végeredményben!

A θ értéke, amely derékszögű rendszerről polárisra való átszámításnál adódik, a $-90..270^\circ$, $-\frac{\pi}{2} \dots \frac{3\pi}{2}$ rad, illetve $-100 \dots 300^\circ$ tartományban van.

A 4. negyedben levő szögértékeket negatív szögekként kezeljük. Az átszámításhoz a szövegységet meg kell választani, hogy azt egyértelműen alkalmazzuk az adatbeírásnál és a kijelzésnél.

Pl. számításuk át derékszögű koordinátákra az $R=5$, $\theta=30^\circ$ poláris koordinátákat! A szövegység: fok.

Billentő	Kijelzés	Megjegyzés
5 $x \approx t$	0	az R beírása
30 2nd P→R	2.5	a θ beírása, az y kijelzése
$x \approx t$	4.330127019	az x értéke

Példa: számítsuk át az $x=3$, $y=4$ pontot poláris koordinátákra!

A szövegység: radián.

Billentő	Kijelzés	Megjegyzés
CLR 2nd Rad	0	szövegység: radián
3 $x \approx t$	0.	az x értékének tárolása
4 INV 2nd P→R	0.927295218	az y beírása, a θ kijelzése radiánban
$x \approx t$	5.	az R értéke.

A koordináta- és a szögátszámítások mindig egy alprogramon belül mennek végbe, és max. 4 befejezetlen műveletet jelentenek. A poláris-derékszögű koordináta-átszámításnak különösen a vektorszámításoknál van jelentősége.

Statistikai számítások

Gyakran kíváncsiak egy változónak egy másikkal való kifejezése, még akkor is, ha ezek között nem ismeretes a pontos függvénykapcsolat. Ezeket a változókat mint adatpontokat, grafikusán ábrázolhatjuk. Az ábrán az ún. független változót az x-tengelyre, a függő változót pedig az y-tengelyre vesszük fel. A kapott pontok halmazát ezután a változó középértékének, szórásának és szórásnégyzetének kiszámításával elemezhetjük. A pontokra lineáris regresszióval egyenes fektethető, és a felhasználó kiszámíthatja ezen egyenes tengelymetszetét és meredekségét. Ebből kiindulva a további pontok interpolálhatók, illetve extrapolálhatók. A korrelációs együttható pedig felvilágosítást arról ad, hogy mennyire illeszkedik az egyenes a pontokhoz.

Adatbeírás

A 2nd Pgm 1 SBR CLR billentőzűs a készülék előkészítése a statisztikai számításokhoz. Ilyenkor a 01..06 adattárolót és a t-tárolót nullára állítjuk. Természetesen ilyenkor be kell tenni a készülékbe egy software-modult, egyébként a 01..06 adattárolót és a t-tárolót kézi billentőzűsessel vagy megfelelő programozással törölni kell.

Az $x \approx t$ billentő. A t-tároló tartalmát felcseréli a kijelzőtároló x tartamával. Statisztikai számításoknál ezzel a billentővel írjuk be a független változókat.

2nd $\Sigma +$ összegzés. Az értékpárokat (x_i, y_i) az 1..6 adattárolóban tároljuk.

Az INV 2nd $\Sigma +$ billentyűzéssel valamely nem kívánt pont, értékpár törölhető. A helytelen beírást úgy töröljük, hogy a nem kívánt értéket újra beírjuk és utána INV 2nd $\Sigma +$ billentyűzést hajtunk végre. A beírt értékpárok számát a készülék minden beírásnál, ill. törlésnél kijelzi. Kétdimenziós statisztikai adatsort (egy-egy pontra vonatkozólag x_i, y_i) a következő billentyűzéssel írjuk be: $i = 1, 2, 3 \dots N$;

$$x_i \approx t \quad y_i \quad 2nd \quad \Sigma +.$$

A beírt pontok számát a készülék minden beírás után jelöli. A nem kívánt értékpár törlésére a hibás x és y értéket újra beírjuk és INV billentyűt közvetlenül a 2nd $\Sigma +$ előtt nyomjuk meg. Az addig beírt értékpárok száma (N) ekkor automatikusan eggyel csökken. Beírás után minden értékpár az alábbiak szerint kerül be a 01..06 adattárolóba:

tároló	tartalom	
01	Σy	} függő változó
02	Σy^2	
03	N	
04	Σx	} független változó
05	Σx^2	
06	Σxy	

A már csoportosított adatokat közvetlenül is beírhatjuk a tárolóba, így az elemzés azonnal elkezdődhet.

A gép a tárolóban összegezi a beírt értéket. Ezeknek a tárolóknak a tartalmát először törölni kell, különben a statisztikai adatok összegzésekor vagy a halmozott összegzéskor hiba keletkezik. Ezért adatbeírás előtt a 2nd Pgm 1 SBR CLR billentyűzéssel állítsuk 0-ra az 1..6 tárolót és a t-tárolót!

Középérték, szórásnégyzet és korrigált szórás

Az adatok (vagy tetszőleges közbesző pont számítása esetén két vagy több értékpár) beírása után a középérték, a korrigált szórás és a szórásnégyzet bármely adatmezőre kiszámítható.

2nd $\bar{x}$ a függő változó (y-mező) középértékének számítása és kijelzése. Fenti billentyűzés után nyomjuk meg az $x \approx t$ billentyűt a független változó (x-mező) középértékének kijelzésére!

INV 2nd $\bar{x}$ a függő változók (y-mező) korrigált eltéréseinek számítása és kijelzése. Ha ezután megnyomjuk az $x \approx t$ billentyűt, akkor a gép a független változók (x-mező) korrigált szórását jelzi ki.

2nd Op 11 a függő változók (y-mező) szórásnégyzetének számítása és kijelzése. Ha ezután megnyomjuk az $x \approx t$ billentyűt, a készülék a független változók (x-mező) szórásnégyzetét jelzi ki.

A készülék a következő egyenleteket alkalmazza:

$$\text{az } x \text{ középértéke: } \bar{x} = \frac{\Sigma x}{N}$$

$$\text{az } y \text{ középértéke: } \bar{y} = \frac{\Sigma y}{N}, \text{ ahol } N \text{ a beírt összes értékpár száma}$$

$$\text{az } x\text{-mező korrigált szórása: } \sigma_x = \sqrt{\frac{\Sigma x^2 - \frac{\Sigma^2 x}{N}}{N - 1}}$$

az y mező korrigált szórása:

$$\sigma_y = \sqrt{\frac{\Sigma y^2 - \frac{\Sigma^2 y}{N}}{N - 1}}$$

az x mező szórásnégyzete:

$$\sigma_x^2 = \frac{\Sigma x^2}{N} - \bar{x}^2$$

az y mező szórásnégyzete:

$$\sigma_y^2 = \frac{\Sigma y^2}{N} - \bar{y}^2$$

Szerencsére a korrigált szórás és a szórásnégyzet számításakor megvan a lehetőség arra, hogy N vagy $N-1$ szerint súlyozzunk. Az N szerinti súlyozás a „maximum likelihood” becslő függvényhez vezet, amelyet általában alapsokaság leírására használnak. $N-1$ szerinti súlyozás a rendszeres hibáktól mentes, becslő függvényhez vezet; ezt általában szűrőpróbákhoz használják. A szórásnégyzet N szerinti súlyozást, a korrigált szórás $N-1$ szerinti súlyozást alkalmazza. A szórásnégyzet számításánál $N-1$ súlyozás esetén a billentyűzés a következő: INV 2nd $\bar{x}$ x^2 és $x \approx t$ x^2 . A szórást N súlyozással INV 2nd Op 11 $\sqrt{x}$ és $x \approx t$ $\sqrt{x}$ billentyűzéssel kapjuk meg. Tekintsük át a következő táblázatban levő billentyűzést!

Billentyűzés

	súlyozás	y-mező	x-mező
középérték		2nd $\bar{x}$	$x \approx t$
szórás	N	INV 2nd Op 11 $\sqrt{x}$	$x \approx t$ $\sqrt{x}$
szórásnégyzet	N	2nd Op 11	$x \approx t$
korrigált szórás	$N-1$	INV 2nd $\bar{x}$	$x \approx t$
korrigált szórásnégyzet	$N-1$	INV 2nd $\bar{x}$ x^2	$x \approx t$ x^2

Egyváltozós adatok esetén az $x \approx t$ billentyűt nem kell használni. Ezt csak a független változó (x-mező) jellemzőinek beírására és kijelzésére használjuk. A 01..06 tárolóra és a t-tárolóra azonban még szükségünk van.

Elemezzük a következő vizsgálati adatokat: 96, 81, 87, 70, 93, 77!

Billentyű	Kijelzés	Megjegyzés
2nd Pgm 1 SBR CLR	0	előkészítés
96 2nd $\Sigma +$	1.	az 1. beírás
81 2nd $\Sigma +$	2.	a 2. beírás
97 2nd $\Sigma +$	3.	a 3. beírás hibás
97 INV 2nd $\Sigma +$	2.	a 3. beírás törlése
87 2nd $\Sigma +$	3.	a 3. beírás helyesen
70 2nd $\Sigma +$	4.	a 4. beírás
93 2nd $\Sigma +$	5.	az 5. beírás
77 2nd $\Sigma +$	6.	a 6. beírás
INV 2nd $\bar{x}$	9.879271228	korrigált szórás
2nd $\bar{x}$	84.	középérték
2nd Op 11	81.33333333	szórásnégyzet
RCL 01	504.	összpontszám

Figyeljünk meg, hogy a korrigált szórást számítottuk ki először; bár ennek számításához a középértékre is szükség volt!

Példa: nagy mennyiségű 100 cm hosszúságú csődarabot rendeltünk. A hosszúság pontosságát és az alakhűséget meg kell vizsgálni. Az utóbbi mérőszáma: $6,0 \pm 0,01$ g/cm. A vizsgálat során 6 szűrőpróba elemzését kell elvégezni.

Szűrőpróba hossz (cm)	1	2	3	4	5	6
tömeg	101,3	103,7	98,6	99,9	97,2	100,1
	609	626	586	594	579	605

Mekkora a kivett minták átlagos tömege? Milyen a minták alakhűsége?

Billentyű	Kijelzés	Megjegyzés
2nd Pgm 1 SBR CLR	0.	a számítás előkészítése
101.3. $x \approx t$	0.	az x_1 beírása
609 2nd $\Sigma +$	1.	az y_1 beírása
103.7 $x \approx t$	102.3	az x_2 beírása
626 2nd $\Sigma +$	2.	az y_2 beírása
98.6 $x \approx t$	104.7	az x_3 beírása
586 2nd $\Sigma +$	3.	az y_3 beírása
99.9 $x \approx t$	99.6	az x_4 beírása
594 2nd $\Sigma +$	4.	az y_4 beírása
97.2 $x \approx t$	100.9	az x_5 beírása
579 2nd $\Sigma +$	5.	az y_5 beírása
100.1 $x \approx t$	98.2	az x_6 beírása
605 2nd $\Sigma +$	6.	az y_6 beírása
2nd $\bar{x}$	599.8333333	az y-mező átlaga (tömeg)
$\div x \approx t$	100.1333333	az x-mező átlaga (hossz)
=	5.990346205	az alakhűség átlaga (g/cm)
INV 2nd $\bar{x}$	17.05774507	a tömeg szórása
$x \approx t$	2.240238083	a hosszúság szórása

A szűrőpróbák átlagos tömege kb. 599,8 gramm. A gép a csöveket átlagban 100,1 cm hosszúságúra vágta. Az alakhűség jobb mint 5.99 g/cm, tehát éppen belül van az előírt tűréshatáron. Végül a különböző darabok tömegének korrigált szórása kb. 17 g, az átlag hossz-eltérése 2,24 cm.

Lineáris regresszió

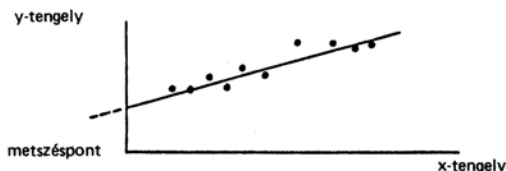
2nd Op 12 az ábrázolt pontokat közelítő egyenesnek az x-tengellyel való metszéspontját számítja és jelzi ki a készülék. Ha a 2nd Op 12 után az $x \approx t$ billentyűt is működtetjük, akkor a közelítő egyenes meredekségét jelzi a gép. Ha az adathalmazt függőleges egyenes jellemzi, akkor szélsőséges esetként az $y \approx t$ tengellyel nincs metszéspont és a meredekség végtelen. A gép ezt meg nem engedett műveletként jelzi. A korrelációs együttható számításakor (2nd Op 13) a készülék függőleges vagy vízszintes egyenest ad meg, és a kijelző villog.

2nd Op 13 az egyes adatpontok korrelációs együtthatójának számítása és kijelzése, a pontokat közelítő egyenesre vonatkoztatva. A korrelációs együttható értéke -1 és $+1$ között van; teljes korreláció esetén ± 1 . Ha a meredekség nulla vagy végtelen, a kijelző villog. Ezt a feltételt a programban 2nd Op 19 billentyűzéssel és programjelzővel lehet vezérelni.

2nd Op 14 egy, a regressziós egyenesen fekvő becsült y' érték számítása és kijelzése, amely egy, a billentyűzettel beírt x értékeknek megfelel. Ha a beírt adatok függőleges egyenest írnak le (végtelen meredekség), akkor az y' értékét nem kell számítani.

2nd Op 15 egy, a regressziós egyenesen fekvő becsült x' érték számítása és kijelzése, amely egy, a billentyűzettel beírt y értékek felel meg. Ha a meredekség 0, a kijelző villog. Az adatbeírás lineáris regresszió esetén azonos a középérték, szórás és szórnégyszert számításnál alkalmazottal; ezek a számítások itt is végrehajthatók. Ha már egyszer beírtunk egy adatsortot, ebből végrehajtható az összes statisztikai művelet. A lineáris regresszió megadja két változó mennyiség közötti összefüggést. A lineáris regresszió számítását a gép a legkisebb négyzetek módszerével végzi, ezen eljárás során a tényleges adatpontoknak a legjobban illeszkedő egyenestől való eltéréseinek négyzete minimális. Gyakorlatilag az történik, mintha felrajzolnánk a pontokat és úgy illeszténénk ezekhez az egyenest, hogy az a pontok között fekdűjk. A pontok és az egyenes közti távolságok négyzetösszege lesz minimális.

Ezt az egyenest az $y = mx + b$ egyenlet írja le, ahol m az egyenes meredeksége, b az y tengellyel való metszéspontja. Mivel az adatok többnyire nem esnek egy vonalba, tisztázni kell, milyen jól közelíti az egyenes az adatokat. Ennek mértéke a korrelációs együttható, mely a változók paramétereiből és a lineáris egyenletből számítható.



A regressziós egyenesnek az y tengellyel való metszéspontját ill. meredekségét a következőképpen határozzuk meg

$$\text{meredekség: } m = \frac{\frac{\sum xy}{N} - \frac{\sum x \sum y}{N^2}}{\frac{\sum x^2}{N} - \frac{(\sum x)^2}{N^2}}$$

$$\text{metszéspont: } b = \frac{\sum y - m \cdot \sum x}{N}$$

$$\text{A korrelációs együttható: } R = \frac{m \cdot \sigma_x}{\sigma_y}$$

A további pontokat ezután előre meg tudjuk határozni. Egyszerűen kiválasztunk egy új y vagy x értéket a regressziós egyenesen. Az eljárás alapja az $y = mx + b$ egyenlet, ahol az m meredekségét és a b metszéspontot a korábban beírt adatok szabják meg.

Pl. egy biztosító társaság megállapítja, hogy a szerződéskötések száma a munkatársak számától függően a következő adatok szerint változik:

az üzletkötők száma	7	12	3	5	11	8
az üzletkötések száma (1000/hó)	99	152	81	98	151	112

Hány üzletkötőnek kell közreműködni, hogy a forgalom havi 200 000 Ft-ra emelkedjen? Milyen havi forgalom várható 13 üzletkötő mellett?

Billentyű	Kijelzés
2nd Pgm 1 SBR CLR	0.
7 x ≈ t	0.
99 2nd Σ +	1.
12 x ≈ t	8.
152 2nd Σ +	2.
3 x ≈ t	13.
81 2nd Σ +	3.
5 x ≈ t	4.
98 2nd Σ +	4.
22 x ≈ t	6.
151 2nd Σ +	5.
22 x ≈ t	23.
151 INV 2nd Σ +	4. }
11 x ≈ t	21.
151 2nd Σ +	5.
8 x ≈ t	12.
112 2nd Σ +	6.
200 2nd Op 15	17.81578947
15 2nd Op 14	176.5561798
2nd Op 12	51.66853933
x ≈ t	8.325842697

Megjegyzés

A számítás bevezetése

az első x érték

az első adatpont

a második x-érték

a második adatpont

stb.

hibás beírás

hibás beírás törlése

a 200 000 Ft-hoz szükséges üzletkötők száma

a forgalom 15 munkatárs esetén

az egyenesnek az y tengellyel való metszéspontja

az egyenes meredeksége

A meredekség és az y tengellyel való metszéspont kiszámítása után felrajzolhatjuk az egyenest. A meredekség az egy üzemeltetőre eső kötések növekedését jelenti, az y tengellyel való metszéspont az üzlet-kötők nélküli forgalom.

Különös jelentősége van annak, hogy a két valószínűségi változóra más matematikai függvényt, azaz regressziótípust alkalmazhatunk. Ha az egyik változó logaritmusa alkalmazzuk adatbeírásakor, akkor egy féllogaritmikus illesztőgörbe jön létre. Ezeket a változatokat a természetes alapú logaritmus-függvény, az exponenciális függvény, a hatvány-függvény, a gyökfüggvény és a reciprokfüggvény alkalmazásával hozhatjuk létre.

Az adatelemzés kezdetén meg kell választani azt a görbétípust, amely leginkább jellemző a feladatban szereplő adatokra. Kísérletezhetünk több görbétípussal is, hogy lássuk, melyik görbe közelíti meg jobban.

Például egy város az alábbi népesedési adatokat közli:

év	1930	1940	1950	1960	1970
lakosság	3221	5361	9212	15410	27612

Mekkora népesedést jósolhatunk 1980-ra, és előreláthatólag melyik évben nő a lakosság 50 000-re?

A népesedési adatokra jellemző, hogy $y = a \cdot e^{b \cdot x}$ alakú exponenciális függvénnyel írható le. Ha az egyenlet mindkét oldalának logaritmusa vesszük, az $\ln y = \ln a + \ln b \cdot x$ alakot kapjuk. Ezért ha a két koordináta-tengelyre x és $\ln y$ értékeit vesszük fel, így egy ún. lin-log összefüggést ábrázolhatunk az egyenessel.

Billentyű	Kijelzés
2nd Pgm 1 SBR CLR	0.
1930 $x \approx t$	0.
3221 $\ln x$ 2nd $\Sigma +$	1.
1940 $x \approx t$	1931.
5361 $\ln x$ 2nd $\Sigma +$	2.
1950 $x \approx t$	1941.
9212 $\ln x$ 2nd $\Sigma +$	3.
1960 $x \approx t$	1951.
15410 $\ln x$ 2nd $\Sigma +$	4.
1970 $x \approx t$	1961.
27612 $\ln x$ 2nd $\Sigma +$	5.
1980 2nd Op 14 INV $\ln x$	46081.80979
50000 $\ln x$ 2nd Op 15	1981.524472

1980-ban a lakosság eléri a 46 080-at, 1981-ben az 50 000-et.

Trendek elemzése

Olyan adatoknál, amelyek egy-egy időperiódusra (évre, napra) vonatkoznak valamely esemény bekövetkezésekor, a gép az x értékét az értékpár beírásnál automatikusan 1-gyel megemeli. Induláskor a készülék a t -tároló pillanatnyi tartalmát mint x értéket hozzárendeli az első adatponthoz, ehhez rendre hozzáad egyet a második, a harmadik stb. adatpont beírásánál. Minden adatpontot a 2nd $\Sigma +$ billentyűzéssel kell beírni. Az x kezdeti értékét az első x érték beírásával egy tetszőleges számra állíthatjuk be, innen kiindulva ezt a gép mindig eggyel növeli: $x_1 x \approx t$, y_1 2nd $\Sigma +$, y_2 2nd $\Sigma +$, y_3 2nd $\Sigma +$ stb. Nem kívánt adatbeírás törölésére nyomjuk meg az $x \approx t - 1$ $x \approx t$ billentyűt, majd írjuk be a nem kívánt y értéket!

Egy számítógépgyártó cég a következő évi nyereségeket közölte:

év	1962	1963	1964	1965–1970	1971	1972	1973	1974
nyereség milliókban	-2,1	0,3	0,8	szünetel	2,9	2,8	3,6	4,0

Milyen nyereséget várhatunk 1980-ban, és mikor lépjük át a 10 milliós határt?

Billentyű	Kijelzés	Megjegyzés
2nd Pgm 1 SBR CLR	0.	a számítás bevezetése
1962 $x \approx t$	0.	az x beállítása
2.1 +/- 2nd $\Sigma +$	1.	az 1962-es veszteség
.3 +/- 2nd $\Sigma +$	2.	az 1963-as veszteség
.8 2nd $\Sigma +$	3.	az 1964-es nyereség
1971 $x \approx t$	1965.	az x újbóli beállítása
2.9 2nd $\Sigma +$	4.	az 1971-es nyereség
2.8 2nd $\Sigma +$	5.	az 1972-es nyereség
3.6 2nd $\Sigma +$	6.	az 1973-as nyereség
4 2nd $\Sigma +$	7.	az 1974-es nyereség
1980 2nd Op 14	6.52181966	
10 2nd Op 15	1988.297788	

A cég 1980-ban 6,5 millió nyereséget vár. A tízmilliós határt előreláthatóan 1988-ban éri el.

A statisztikai számításoknál bármelyik statisztikai mutatószámot alkalmazhatjuk. Így az utóbbi példában ugyanígy meghatározhatjuk a középértéket, a szórást, a szórásnégyzetet, a meredekséget és a tengelymetszetet.

Statisztika a számításokban

Statisztikai számításokat összetettebb műveletek közben is végezhetünk. Számítás közben legfeljebb 4 művelet maradhat befejezetlen, ekkor még közbeiktathatunk statisztikai számításokat is. Ezekhez a gépen belül legfeljebb 4 feldolgozási „síkot” vehetünk igénybe.

Példa: egy biztosítótársaság üzemviteli költsége a $3 + 2 \cdot 1,2^{(4+N)}$ képlettel számítható.

A képletben N az üzletkötők száma, akiket 200 000 Ft forgalom elérésére alkalmaztak. Billentyűzzük be a $3 + 2 \cdot 1,2^{(4+)}$ kifejezést, és az utolsó példákhoz hasonlóan számítsunk lineáris regressziót! Miután a megadott forgalom eléréséhez szükséges munkatársak számát meghatároztuk, zárjuk le a számítást a = utasítással! Ügyeljünk arra, hogy a statisztikai számítások során 4 művelet maradt befejezetlen.

Egy program részét képező statisztikai számítások egy alprogramsíkot igényelnek. Erről részletesebb ismereteket az alprogramoknál adunk.

A PROGRAMOZÁSRÓL ÁLTALÁBAN

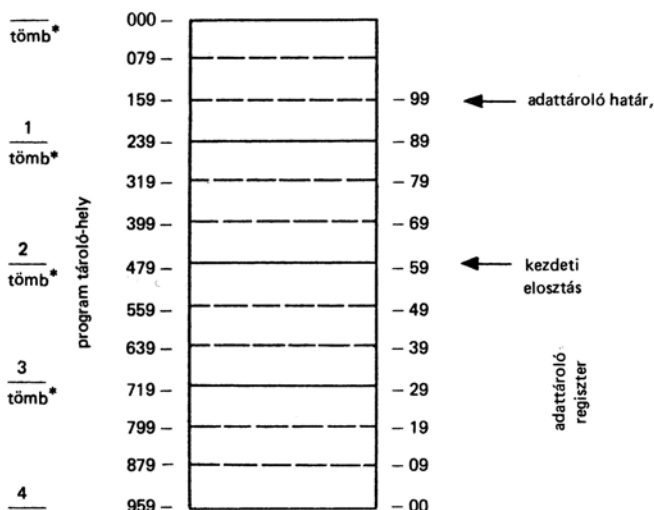
A számológép programozása

Ha egy problémát kézi billentyűzéssel akarunk megoldani, először határozzuk meg a számításához szükséges műveletek és függvények sorrendjét, és csak utána billentyűzzük be a programlépéseket! A programozás nem jelent sokkal többet, mint a tanulási üzemmód beírása, továbbá a gépnek adott utasítást a billentyűzés megjegyzésére. Valójában az egyes billentyűzési utasítások a programtároló helyekre jutnak, így minden billentyűzés programutasítássá válik. Így épül fel a program. Ha a programban foglalt utasításokat végrehajtjuk, ugyanazt az eredményt kapjuk, mint a készülék billentyűzésekor. Ha a programot egyszer már tároltuk, az ismét végrehajtható új változókkal, ezáltal főlélegessé válik a programutasítások ismételt beírása. Így nemcsak időt takarítunk meg a beírásnál, hanem csökkentjük a beírási hibákat is. A programot a programtároló addig tárolja, míg másikkal nem helyettesítjük, azaz amíg a 2nd CP billentyűzéssel nem töröljük, vagy amíg a készüléket ki nem kapcsoljuk. A közbeni időben a programot tetszés szerint alkalmazhatjuk. Eredményhez juthatunk pl. valamely tárolt program útján is, de közbeiktathatunk egy sor kézi billentyűzésű műveletet. Máskor a programot egyszerűen előhívjuk, végrehajtjuk, és az eredménnyel folytatjuk a kézi billentyűzésű számításokat.

A készülék programozási rendszere alkalmazkodik a legmagasabb követelményekhez, és alkalmazása mégis egyszerű. A számítások felépítésének alapos ismeretében olyan készülék birtokában vagyunk, amely a lehetőségek tekintetében igen közel áll a számítógéphez.

Tárolókapacitás és annak elosztása

A készüléken belüli teljes tárolómező adat- és programtárolásra szolgál.



*tömb: közös tárolómező, adatok és programlépések számára.

A teljes tárolómező 120 regisztert tartalmaz, melyet az ábrán négy tömbbel ábrázoltunk. Egy-egy regiszter 8 programutasítás tárolására alkalmas, így a tárolómező teljes kapacitása 960 tárolóhely. Ha egyenlő arányban osztjuk fel a tárolómezőt az adatok és a programutasítások között, akkor 480 program-lépés, azaz 60-60 regiszter lesz az elosztási arány.

Ezt az elosztást 10 regiszteres növekményekkel módosíthatjuk: pl. használhatunk 120 regisztert egy programban és ez esetben nem marad hely az adattárolásra vagy lehet 100 adatregiszterünk és $20 \times 8 = 160$ programutasításunk. De választhatunk bármely más kombinációt is, az egyetlen megkötés a rendszer részéről, hogy maximálisan 100 adatregiszter alkalmazható, mivel minden program-tárolási művelet egy kétjegyű címet kíván, így a 00..99 tartomány az adattárolás maximális tartománya.

A tárolómező elosztása céljából írjuk be az adattárolók szükséges tízes csoportjainak számát (0...09), és billentyűzzük 2nd Op 17-et, pl. 20 adattároló esetén 2 2nd Op 17-et! A kijelző az elosztást így jelzi:



Ez a kijelzés azt jelenti, hogy adattárolásra 20 tárolóregiszter (00..19) szolgál, a programtároló mező 800 tárolóhelyből áll, a 000...799 címmel. A 2nd Op 16 billentyűzéssel a pillanatnyi elosztást mindenkor ellenőrizhetjük, és a gép a fenti módon kijelzi.

Alapvető programvezérlési utasítások

A fontosabb vezérlési utasításokat már ismerjük, így hozzáfoghatunk gépünk programozásához.

LRN (learn): ez billentyű megnyomásakor a készülék tanulási üzemmódba kapcsol. Elkezdhetjük a programtároló lefoglalását. Az LRN ismételt lenyomásakor a vezérlést visszaadjuk a billentyűzetnek. Ilyenkor visszaáll az eredeti kijelzési mód. Az LRN billentyű alkalmazásakor a legutóbbi érték megmarad a kijelzőben de a kijelzés esetleges villogása leáll. A tanulási üzemmódot nem írhatjuk be, ha a programtárolóban védett program van.

2nd CP (clear program): ezzel töröljük a programtároló összes helyét, az alprogram-visszaléptető tárolót, a t-tárolót; ezenkívül az összes jelzőt (-flaget) kiiktatjuk, és a programmutatót 000 helyzetbe állítjuk. Ilyenkor az adatvédelem is megszűnik. Ugyanez az utasítás egy program részeként csak a t-tárolót nullázza.

R/S (run/stop): pillanatnyi üzemmód-váltás.

Az R/S billentyűzésekor a program végrehajtása megszakad. Megszakított program esetén a következő R/S utasítás hatására a programmutató pillanatnyi helyzetének megfelelően folytatódik a végrehajtás. Így a teljes program befejezése előtt beírhatunk vagy megvizsgálhatunk közbeni adatokat is.

RST (reset): ez az utasítás a programmutatót a tároló 000 helyére állítja vissza, ami a software-programra is érvényes. Az RST törli az alprogram-visszaléptető tárolót, és kiiktatja a programjelzőket. Ezenkívül ezt az utasítást használjuk a software-program megszakítására is, ha a program önmagát nem képes leállítani. Ebben az esetben viszont minden eredmény elvész.

2nd Pause: a program végrehajtása közben a készülék a kijelzőtároló tartalmát kb. 0,5 másodpercig kijelzi. Szünetutasítást szükség szerint folyamatosan is adhatunk. Ha tehát a billentyűt a program futása közben nyomjuk meg, akkor a készülék az összes lépést kijelzi. Software-programok és védett programok futása közben ez a billentyűzés hatástalan.

Learn üzemmód

Ha rögzítettük a számítás sorrendjét, kapcsoljuk a készüléket tanulási üzemmódba (2nd CP LRN), majd a rögzített számítási sorrendet billentyűzzük be a programtárolóba! A 2nd CP utasítással biztosítjuk, hogy a 000 tárolóhellyel kezdődő program beíródjék. A tanulási üzemmódban a kijelzés alakja a következő.



A programtároló helyek 000-val kezdődnek, és a választott elosztásnak megfelelő folyamatos számozást kapnak. Kezdetben a programtároló tartalma az összes helyen nulla. A tartalom mindaddig nulla marad, amíg azt meg nem változtatjuk.

Általában minden tárolóhely egy műveletet, egy címet vagy csak egy számjegyet tartalmaz.

Az utasításkód (billentyűkód) egy kétjegyű szám, amelyet az egyes műveleteknek megfelelően az egyes billentyűknek a billentyűzetben elfoglalt tényleges helyzete határoz meg. Ha egy programot be akarunk írni, a gép általános esetben 00 billentyűkódot mutat. Ez azért van, mert a készülék egy utasítás teljes beírása után automatikusan továbblép a következő rendelkezésre álló tárolóhelyre. Az utasításkódokról részletes ismertetést a programszerkesztéssel kapcsolatos fejezetben találhatunk.

A programmutató (utasításcím-tároló) megadja az ellenőrzés lehetőségét; megmutatja, hogy a programtárolóban melyik címhez értünk el. Tanulási üzemmódban a mutató végigfut a programtárolón és a készülék mindig a következő tárolóhelyet jelzi ki.

A programnak a programtárolóba való bebillentyűzése után újra megnyomjuk az LRN billentyűt, hogy ezzel a gépet ismét billentyűvezérlésre kapcsoljuk. Majd beírjuk a változókat, és elindítjuk a program végrehajtását.

Saját program beírása a következőképpen történik:

1. Az RST vagy 2nd CP utasítást billentyűzzük. Bármelyikük hatására az utasításcím-tároló 000-ra, a programtároló az első tárolóhelyére áll be.
2. Az LRN billentyűvel tanulási üzemmódba kapcsolunk. Az ötjegyű kijelzési mód jellemzi ezt az üzemmódot.
3. Bebillentyűzzük a teljes programot lépésenként, beleértve a 2nd és INV előválasztó utasításokat is. A kijelző mindig a következő szabad tárolóhelyet mutatja.
4. Ügyeljünk arra, hogy a program a programtároló kapacitását ne lépje túl! Ha az utolsó tárolóhelyet is betöltöttük, a készülék automatikusan visszakapcsol kézi üzemmódba, és megváltozik a kijelzési mód.
5. Újra nyomjuk meg LRN billentyűt a tanulási üzemmód megszüntetésére!
6. Futtassunk le egy ismert eredményű vizsgálatot a program helyességének ellenőrzésére és szükség esetén módosítsuk a programot!

A következő példa szemlélteti a fent leírtakat. Készítsünk programot egy r sugarú, h magasságú henger térfogatának kiszámítására.

Az egyenlet: $V = \pi r^2 h$.

A szükséges programműveletek: az r értékének beírása, a program elindítása, leállítás a h értékének beírásához, a térfogat kiszámítása, leállítás és eredménykijelzés.

Billentyűzés	Kijelzés	Megjegyzés
2nd CP	0	a programmutató beállítása a 000 tárolóhelyre, a programtároló törlése
LRN	000 00	a gép átkapcsolása tanulási üzemmódba, a program-beírás kezdete
x^2	001 00	r^2 (az r értékét előzőleg beírtuk)
X	002 00	
2nd π	003 00	csak 1 tárolóhelyet foglalunk le
X	004 00	$\pi \cdot r^2$ a kijelző-tárolóban
R/S	005 00	leállítás a h értékének beírásához
=	006 00	az eredmény kiszámítása
R/S	007 00	a program leállítása, a V kijelzése
RST	008 00	visszatérés a 000 helyen levő utasításhoz
LRN	0	visszatérés billentyűvezérlésre

Az utolsó beírt billentyűutasításnak különleges szerepe van. Az R/S billentyű megszakítja a végrehajtást és kijelzi az eredményt. Az RST utasítással végrehajtjuk az automatikus visszatérést a 000 tárolóhelyre, ha az r értékét egy új változócsoporthoz írjuk be, és az R/S billentyűt megnyomjuk.

Az R/S billentyűvel megfordítjuk a program pillanatnyi feldolgozási helyzetét. Ha ezt a billentyűt megnyomjuk, vagy ha az utasítás a végrehajtás közben programutasításként szerepel, akkor a feldolgozás megszakad, és a készülék átkapcsol billentyűvezérlésre. Ha az R/S utasítást kézzel billentyűzve adjuk be, a programfutás újra megindul, és a számítás folytatódik, ahol abbamaradt. A programmutató ott marad, ahol a program megszakításának pillanatában áll. Tehát a végrehajtás a számítások zavarása nélkül folyhat, feltéve, hogy nem hajtottunk végre változtatásokat.

Saját program végrehajtása

Egy program végrehajtása során a készülék az utasításokat – kezdve a programmutató pillanatnyi címével – a megfelelő sorrendben hajtja végre. A kivételeket később tárgyaljuk. Ha a készülék a végrehajtást a programhatáron túl is folytatná, a feldolgozás megszakad és a kijelző villog. Ezért a programokat az R/S, INV, SBR vagy a később tárgyalandó átmeneti ill. műveleti utasításokkal kell befejezni. A programtárolóban levő térfogatszámítás programja végrehajtható.

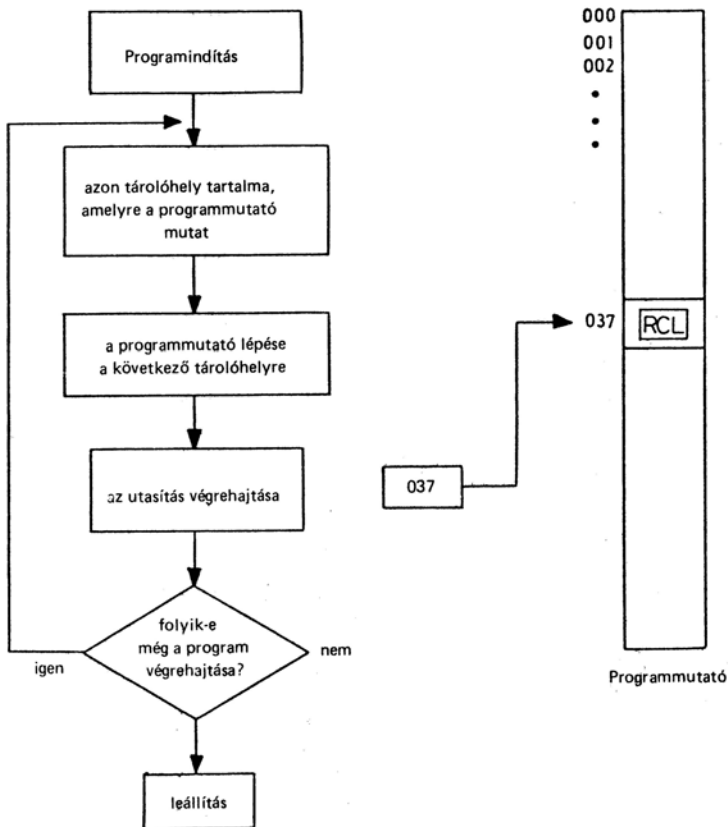
Számítsuk ki a térfogatot, ha $r = 3$, és $h = 9$!

Billentyű	Kijelzés
RST	0
3	3
R/S	(sötét)
	28.27433388
9	9
R/S	(sötét)
	254.4690049

Megjegyzés
 a programmutató beállítása 000-ra
 az r érték beírása
 a program végrehajtásának kezdete
 $\pi \cdot r^2$ értéke a kijelzőtárolóban
 a h értékének beírása
 a feldolgozás folytatása
 a program leállítása, a V értékének kijelzése

Figyeljük meg, hogy a program végrehajtása közben a kijelző sötét marad, csak egy gyengén világító „C” látható a kijelző bal sarkában! Az az idő, amíg a kijelző sötét, erősen változhat aszerint, hogy melyik program fut és hogy milyen táplálásra kapcsoltuk a készüléket.

Ha végrehajtottunk egy utasítást, a programmutató folyamatosan ellenőrzi, és mindig arra az utasításra mutat, amelyet éppen végre kell hajtani.



Mivel további programozási lehetőségeket is tárgyalunk a programmutató szerepkörét is bővítjük.

Programozási műveletek

SST (single-step): a programmutató egy lépéssel előre lép. Programozási üzemmódban a billentyűnek hatására a készülék kijelzi a következő tárolóhelyet. Ha az SST utasítást kézzel billentyűzzük, a készülék a programot lépésenként hajtja végre, miközben minden egyes lépést kijelez.

BST (back-step): A tanulási üzemmódban a billentyű hatására a programmutató egy lépéssel visszalép, és az ott tárolt utasításkódot jelzi ki.

Software-program végrehajtásakor ez a két billentyűzés hatástalan.

A programok billentyűzésére, és a már bevitt programok ellenőrzésére két utasítás van, ezek a programon belüli léptetést is lehetővé teszik. Ha tanulási üzemmódban megnyomjuk az SST léptető billentyűt, a programmutató eggyel továbblép, a készülék a következő tárolóhely és utasításkód számát a tárolt utasítás zavarása nélkül kijelzi. Az SST utasítás ismételt beadása tehát a programtárolón való lépésenkénti átfutást és az itt tárolt utasítások felülvizsgálatát biztosítja. Az egyszerű lépés utasítását programvégrehajtási üzemmódban is beadhatjuk. Billentyűzés esetén az SST utasítás a tárolt program lépésenkénti tényleges végrehajtását biztosítja. Valahányszor az SST billentyűt benyomjuk, a készülék a programmutató pillanatnyi helyzetének megfelelően hajtja végre az utasítást, és a programmutató továbblép, mint a program végrehajtásakor. A kijelző az utasítás alapján számított értéket jelzi ki. Néha több egyszerű léptetésre van szükség, mivel a folyó művelet gyakran több lépésből áll. A +RCL 09 utasításnál 3 egyszerű léptetési utasításra van szükség, a 09 tároló tartalmának előhívására.

Ha a programot tanulási üzemmódban egyenkénti előre, vagy hátralépéssel ellenőrizzük, a különböző utasításokat tároló lépések billentyűkódjai megjelennek.

Utasításkódok (billentyűkódok)

Program üzemmódban a programmutató pillanatnyi helyzetét és az illető tárolóhelyhez tartozó utasítást jelzi a gép. Az utasítást kétjegyű kóddal jellemezzük, amely a billentyűnek a billentyűzetben elfoglalt helyzetéből adódik. Az első számjegy (1...9) felülről lefelé adja meg a 9 sor egyikét, amelyben a billentyű található. A második számjegyből következtetünk az oszlopra, balról jobbra (1...5). Pl. az STO billentyű a 4. sorban és a 2. oszlopban található, utasításkódja 42. A másodlagos rendeltetésű billentyűnél a meghatározott sor-oszlop címhez ötöt kell hozzáadni. Azaz 2nd sin esetén a kód 38 lesz, mert 33-mal az x^2 függvényt jellemeztük. A jobb szélső oszlop esetén a másodfüggvényekhez úgy kell ötöt hozzáadni, hogy a sorok száma ne változzék, pl. 2nd D'-nél a kód 10, és nem $15+5=20$. A 0..9 számjegybillentyűhöz a 00..09 kódokat rendeljük.

Billentyűzet és billentyűkódok

Billentyű	Billentyű-kód	Billentyű	Billentyű-kód	Billentyű	Billentyű-kód	Billentyű	Billentyű-kód	Billentyű	Billentyű-kód
A'	16	B'	17	C'	18	D'	19	E'	10
A	11	B	12	C	13	D	14	E	15
2nd	összevont utasítás	INV	27	log	28	CP	29	CLR	20
		INV	22	lnx	23	CE	24	CLR	25
Pgm	36*	P→R	37	sin	38	cos	39	tan	30
LRN	–	x↔t	32	x²	33	√x	34	1/x	35
Ins	–	CMs	47	Exc	48*	Prd	49*	Ind	40 vagy összevont utasítás
SST	–	STO	42*	RCL	43*	SUM	44*	y^x	45
Del	–	Eng	57	Fix	58*	Int	59	x	50
BST	–	EE	52	(	53	)	54	÷	55
Pause	66	x=t	67*	Nop	68	Op	69*	Deg	60
GTO	61*	7	07	.8	08	9	09	X	65
Lbl	76*	x≥t	77*	Σ+	78	$\bar{x}$	79	Rad	70
SBR	71*	4	04	5	05	6	06	–	75
St flg	86*	If flg	87*	D.MS	88	π	89	Grad	80
RST	81	1	01	2	02	3	03	+	85
Write	96	Dsz	97*	Adv	98	Prt	99	List	90
R/S	91	0	00	•	93	+/-	94	=	95

A * jelű utasítások csak más utasításokkal, címekkel teljesek. Az Ind utasítást bizonyos esetekben, a 2nd utasítást mindig annak a billentyűnek a kódjával vonjuk össze, amellyel együtt alkalmazzuk.

A készülékhez átlátszó fóliából készült feltétet adunk, amely a billentyűk fölé helyezve jelzi az egyes billentyűkhöz tartozó billentyűkódokat.

Billentyűkód	Billentyű	Billentyűkód	Billentyű	Billentyűkód	Billentyű
00	0	39	2nd cos	72	STO 2nd Ind
↓	↓	40	2nd Ind	73	RCL 2nd Ind
09	9	42	STO	74	SUM 2nd Ind
10	2nd E'	43	RCL	75	—
11	A	44	SUM	76	2nd Lbl
12	B	45	γ^x	77	2nd $x \geq t$
13	C	47	2nd CMs	78	2nd $\Sigma +$
14	D	48	2nd Exc	79	2nd $\bar{x}$
15	E	49	2nd Prd	80	2nd Grad
16	2nd A'	50	2nd x	81	RST
17	2nd B'	52	EE	83	GTO 2nd Ind
18	2nd C'	53	(	84	2nd Op 2nd Ind
19	2nd D'	54	)	85	+
20	2nd CLR	55	$\div$	86	2nd St flg
22	INV	57	2nd Eng	87	2nd If flg
23	Inx	58	2nd Fix	88	2nd D.MS
24	CE	59	2nd Int	89	2nd π
25	CLR	60	2nd Deg	90	2nd List
27	2nd INV	61	GTO	91	R/S
28	2nd log	62	2nd Pgm 2nd Ind	92	INV SBR
29	2nd CP	63	2nd Ext 2nd Ind	93	.
30	2nd tan	64	2nd Prd 2nd Ind	94	+/-
32	$x \leftrightarrow t$	65	X	95	=
33	x^2	66	2nd Pause	96	2nd Write
34	$\sqrt{x}$	67	2nd $x = t$	97	2nd Dsz
35	1/x	68	2nd Nop	98	2nd Adv
36	2nd Pgm	69	2nd Op	99	2nd Prt
37	2nd P → R	70	2nd Rad		
38	2nd sin	71	SBR		

A készülék használata mellett hamar megtanuljuk a leggyakrabban előforduló billentyűkódokat, ezért a fenti táblázatra nem lesz szükség. A legtöbb kódot a billentyűzetről, esetleg a fóliáról olvashatjuk le.

A billentyűutasítások tárolása

A legtöbb utasítás egyetlen helyet foglal el a programtárolóban. Néhány utasításhoz azonban több billentyű kezelése szükséges egyetlen programtárolóhely esetén is. Az adattárolási utasítások, a software-programok kezelése és a különleges vezérlési műveletek után álló kétjegyű címek egységet képeznek, és csak egyetlen tárolóhelyet foglalnak el. Pl. az RCL 16 csak két tárolóhelyet foglal el, hasonlóan a 2nd Pgm 12-höz, ahol 2nd Pgm az egyik, a 12 a következő tárolóhelyre kerül. Ezt a műveletet a készülék automatikusan végzi.

A szükségtelen átmeneti címeket, (pl. a GTO 123) a következőképpen tároljuk: egy tárolóhely a GTO utasításnak, egy a 01 számnak és egy a következő 23-nak. Ezen utasítás billentyűzésekor a gép a billentyű-utasításokat minden külön beavatkozás nélkül elhelyezi a megfelelő tárolóhelyeken.

Ha az Ind billentyűt más utasítással együtt alkalmazzuk, az összevont utasítás csak egyetlen tárolóhelyet foglal el, és más utasításkódot kap. Az új kódot nem az Ind billentyű helye határozza meg, hanem a hozzárendelt billentyű. Az STO 2nd Ind utasításnak pl. 72 a kódja és csak egy tárolóhelyet foglal el. Ezt a különleges hozzárendelést a készülék végzi el. A hozzárendelést a billentyűkódok számszerű sorrendjében közölt táblázat tartalmazza. Ez az alábbi közvetett utasításokat érinti:

Billentyű	Billentyűkód
2nd Pgm 2nd Ind	62
2nd Exc 2nd Ind	63
2nd Prd 2nd Ind	64
STO 2nd Ind	72
RCL 2nd Ind	73
SUM 2nd Ind	74
GTO 2nd Ind	83
2nd Op 2nd Ind	84

Azok a közvetett utasítások, amelyeket az összevonás nem befolyásol, tárolják a 2nd Ind utasításkódját, mégpedig az Ind utasításnak a billentyűzetben elfoglalt helye szerint. Pl. a $2nd\ x = t$ 2nd Ind tárolása: 67 kód az egyik tárolóhelyen, (2nd $x=t$) 40 kód a következőben (2nd Ind). A közvetett utasítások alkalmazásának lehetőségeit később még tárgyaljuk.

A PROGRAMOK MÓDOSÍTÁSA

2nd Nop (no-operation): program üzemmódban ezzel a billentyűvel törölhetjük a nem kívánt utasítást, illetve a programlépések között későbbi kiegészítések céljára lépések helyét hagyhatjuk szabadon. A program végrehajtásakor ezeken a helyeken nem jön létre művelet. Ennek a billentyűnek az alkalmazása nem befolyásolja a billentyűzést, az utasítást és az adatbeírást, vagyis a billentyűt címként használjuk.

2nd Del (delete): egyes utasítások törlése. Program üzemmódban a kijelzett utasítást töröljük, és minden következő utasítás egygel előbbre ugrik, így a hiány megszűnik. A programmutató ennél a műveletnél nem mozog.

2nd Ins (insert). Program üzemmódban a kijelzett utasítástól kezdődően az összes utasítás eltolódik egy hellyel a programtároló vége felé. Így a beiktatandó utasításnak teremtünk helyet.

A programtárolóban levő program átdolgozásának lehetőségei

1. a program lépésenkénti végrehajtása előre és hátra, miközben az egyes tárolóhelyeken levő utasításokat kijelezzük,
2. egyik utasítás helyettesítése másikkal,
3. helybiztosítás az új utasítás beiktatásához,
4. a kijelzett utasítás törlése és a hiány kitöltése.

Egyik utasítás helyettesítése másikkal

Tanulási üzemmódban egy billentyűs beírás átírja az előző, ugyanezen címre írt programutasítást. Ha lépésenként ellenőrizzük a programot, és nem kívánt utasítást találunk, akkor a javítás céljából nyomjuk meg a kívánt helyes billentyűt! A készülék a kijelzett helytelen utasítást a helyesre cseréli fel.

Egyetlen utasítás törlése

Egyetlen utasítást a 2nd Del billentyűvel iktathatunk ki a programból. Tanulási üzemmódban törölhetjük a kijelzett utasítást, mire az összes utána következő egység előbbre ugrik, így az üres lépés helye megszűnik. Ennek az a következménye, hogy a programtároló utolsó helye felszabadul, és egy nulla foglalja el. Így a programnak tetszőleges terjedelmű részét is törölhetjük, ha szükséges.

Egyetlen utasítás beiktatása

Ha a programba még egy utasítást kell beiktatni, nyomjuk meg a 2nd Ins billentyűt! A kijelzett utasítással

kezdődően minden további utasítást egy tárolóhellyel eltolunk, hogy helyet teremtsünk az új utasításnak. Most már csak be kell billentyűzni az új utasítást. Ez a művelet tetszőlegesen ismételhető, és így a programhoz nagyobb részeket is hozzáadhatunk. Ügyeljünk azonban arra, hogy valahányszor végrehajtjuk ezt a műveletet, a programtároló utolsó helyén levő utasítás mindig elvész, továbbá az átmenetek címének számjegyét át kell számolni (pl. GTO 124 helyett GTO 125)!

A 2nd Nop billentyűzéssel törölhető a tárolóhelyen levő utasítás, és egy üres utasítás kódjával helyettesíthető. Ezt a kódot a gép a végrehajtás során egyszerűen nem veszi figyelembe, vagyis azt címkeként tárolja. Ennek a billentyűnek az az alkalmazási lehetősége, hogy vele a programnak egy bizonyos, gyakran használt helye lefoglalható. Ezáltal a programban beiktatás és törlés nélkül is végrehajtható a változtatás.

Például az $x^2 + 3 \cdot x - 2$ kifejezést úgy kell beírni és előkészíteni, hogy az különböző x-értékekre ismételtlen megoldható legyen.

X a végrehajtás előtt legyen a kijelzőben!

Billentyű	Kijelzés	Megjegyzés
LRN	000 00	a tanulási üzemmód beírása
STO	001 00	az x értékének tárolása a későbbi alkalmazáshoz
1	001 00	
x^2	003 00	
—	004 00	hibás beírás
BST	003 75	egyszeri visszalépés 003-ra
+	004 00	a beírás törlése
4	005 00	hibás beírás
X	006 00	
BST	005 65	egyenkénti visszalépés a hibás beíráshoz
BST	004 04	
3	005 65	a 4 helyettesítése 3-mal
SST	006 00	egyszeri előreléptetés a helyes x tárolóhelyig
X	007 00	
RCL	008 00	
1	008 00	
BST	008 01	
BST	007 43	észrevesszük, hogy a X műveletet kétszer írtunk be
BST	006 65	egyedi visszaléptetés a 006 tárolóhelyre
2nd Del	006 43	a X törlése, az RCL eltolása 007-ről 006-ra, stb.
SST	007 01	egyenkénti előreléptetés az RCL szerinti tárolóhelyig
SST	008 00	egyenkénti előreléptetés az 1 szerinti tárolóhelyig
2	009 00	
=	010 00	
BST	009 95	észrevesszünk egy beírási hiányt
BST	008 02	egyedi visszaléptetés a 008 tárolóhelyre
2nd Ins	008 00	a 008 tárolóhely felszabadul, a 2= továbblép
—	009 02	a — jel beiktatása
SST	010 95	egyenkénti előreléptetés a 2 beírása mögé
SST	011 00	egyenkénti előreléptetés az = beírása mögé
R/S	012 00	programleállítás, eredménykijelzés
RST	013 00	
LRN	0	

A program helyes beírásának ellenőrzésére menjünk végig még egyszer lépésenként! A teljes program a következő:

Tárolóhely és billentyűkód	Billentyű
000 42	STO
001 01	0 1
002 33	x^2
003 85	+
004 03	3
005 65	X
006 43	RCL
007 01	0 1
008 75	-
009 02	2
010 95	=
011 91	R/S
012 81	RST

A javítás során az összevont billentyűkódok miatt különösen figyeljünk! Figyeljük meg a következő billentyűzést!

Tárolóhely és billentyűkód	Billentyű
019 95	=
020 42	STO
021 12	1 2
022 61	GTO

Az STO 12 utasítást STO 13-ra kell módosítani. Két megoldás van: vagy a 020 tárolóhelyre lépünk vissza és STO 13 utasítást billentyűzünk, vagy a 021 tárolóhelyre lépünk vissza és C utasítást (billentyűkódja 13) billentyűzünk. Nem megoldás az, ha a 021 tárolóhelyre lépünk vissza és 13-at billentyűzünk, mert az ebben az esetben két programtárolóhelyet foglalna el.

A PROGRAMRÉSZEK MEGJELÖLÉSE

2nd Lbl (label): programszakaszok jelölése csak a tanulási üzemmódban lehetséges. A készüléket ezzel a billentyűvel utasítjuk, hogy a következő billentyűzést jelölésként használja, és ne végrehajtandó utasításként. A címkék alkalmazása a programban a jegyzetfüzetben használt címkékhez hasonlítható. Használatuk különösen többszakaszú programoknál előnyös, amelyeket egyenként vagy egymásután hajthatunk végre. Két típusuk van: az általános címke, és a programcímke.

A programcím-billentyűk mint címkék

A billentyűzet felső billentyűsorát (A...E és 2nd A' ...2nd E') nevezzük programcím-billentyűnek. Egy programszakasznak a fenti billentyűkkel való megjelölése (pontosabban ezek egyikével való megjelölése) lehetővé teszi, hogy számítási üzemmódban a billentyűkkel közvetlenül hozzáférhessünk a megjelölt programszakaszhoz. Tehát lényegében magunk határozzuk meg ennek a billentyűnek további rendeltetését. A megjelölt rész teljesen kitöltheti a programtárolót, de lehet olyan rövid is, mint az előző példában. Ha ezt a példát tárolni akarjuk, adjunk a programnak címkét!

Billentyű	Kijelzés	Megjegyzés
LRN	0	visszatérés a számítási üzemmódra
RST	0	visszaállítás a 000-ra
LRN	000 42	a tanulási üzemmód beírása
2nd Ins	000 00	továbblépés 2 tárolóhelyre
2nd Ins	000 00	
2nd Lbl	001 42	az Lbl elhelyezése a 000-ban
A	002 42	az A elhelyezése a 001-ben

A programot az A címkével láttuk el, így az bármikor végrehajtható. Ekkor egy x értéket írunk be a kijelzőbe, majd megnyomjuk az A billentyűt.

Billentyű	Kijelzés	Megjegyzés
1 A	2.	a kijelzés kiszámítása $x = 1$ értéknél
1 +/- A	-4.	számítás $x = -1$ értéknél
6 A	52	számítás $x = 6$ értéknél
3.2 EE	3.2 00	
12 A	1.024 25	számítás $x = 3,2 \cdot 10^{12}$ értéknél
2nd π ÷	3.1415927 00	
.03 = A	1.1278386 04	számítás $x = \pi : 0,03$ értéknél

A programcím-billentyűt hozzárendelhetjük a program egyik utasítás-sorozatához, amelyet azután tetszőszerint hajthatunk végre. Teljesen mindegy, hogy egy programcím-billentyűt nyomunk meg, vagy az a végrehajtás közben fordul elő; mindkét esetben rááll a gép a program azon részére, amelyet a programcím-billentyűvel megjelöltünk.

Általános címkek

Csaknem az összes billentyűt — beleértve a kettős rendeltetésűeket is — használhatjuk címkeként. Csak a következő billentyűk kivételek: 2nd, LRN, Ins, Del, SST, BST, Ind és a 0...9 számjegyek. Kerüljük továbbá az R/S címkeként való használatát, mert ezzel az utasítással a végrehajtást indítjuk! A fenti címkek hozzárendelése hasonló a programcímkekéhez. Azonban az általános címkekkel megjelölt programszakaszokhoz billentyűzéssel nem olyan egyszerű hozzáférni, mint a programcímkek esetén. Megjelölhetünk pl. az egyik szakaszt a cos-billentyűvel. Billentyűzzünk 2nd cos utasítást, ekkor a gép végrehajtja az utasítást, és a kijelzőben a kijelzett szám koszinusza jelenik meg. Az általános címkek elsődleges rendeltetése a programon belül az átmeneti műveletekkel kapcsolatos; még akkor is, ha a billentyűs hozzáférés lehetséges.

A GTO 2nd cos utasítás pl. a programszámlálót a cos címke utáni utasításra irányítja. Az SBR 2nd cos hasonlóképpen; a feldolgozás éppen ott folytatódik, ahol a GTO 2nd cos R/S billentyűzéssel folytatódott volna.

Általában egy programot annyi címkével lehet ellátni (72 címke áll rendelkezésre), amennyivel szükséges. Minden címke csakis egy programszakaszt jelölhet, de bármilyen gyakran előhívható.

A sornyomatóval felsorolást készíthetünk az éppen tárolt program összes címkeiről, beleértve a billentyű-kódjaikat és a programtárolóban levő címkeiket. Nyomjuk meg a GTO 0-t, vagy az RST-t a programszámláló 000-ra állítása végett, majd a 2nd Op 10-t! Ekkor a készülék kinyomtatja a táblázatot.

ÁTMENETI UTASÍTÁSOK

Ha egy programot végrehajtunk, az eddigiek szerint a feldolgozás a 000 tárolóhelyre kezdődik, és a következő R/S utasításig folytatódik a sorrendek betartásával. A merev feldolgozási sorrend az ún. átmeneti utasításokkal megszakítható. Van feltételes és feltétel nélküli átmeneti utasítás. A feltétel nélküli átmeneti utasítások a programszámlálót azonnal a kívánt tárolóhelyre tolják el. Feltételes átmenetnél először megvizsgáljuk a program állapotát, és az átmenet csak bizonyos, meghatározott feltételek mellett jöhet létre.

A feltétel nélküli átmeneti utasítás és az alprogram utasítás

GTO N vagy nnn: egy programban a „go-to” utasítás a feldolgozási folyamatot azonnal az N címére vagy az nnn tárolóhelyre irányítja. N lehet általános vagy programcímke. Ha „go-to” utasítást billentyűzettel írjuk be, akkor a programszámlálót az nnn tárolóhelyre vagy az N programszakaszra irányítjuk, de a program végrehajtása még nem folytatódik. Az utasítás kézi billentyűzése mindenképp előtti akkor előnyös, ha egy programrészhez módosítási vagy egyéb célból gyorsan hozzáférni akarunk. A „go-to” utasításnak van még egy másik szerepe is, amelyet a függelékben részletezünk. Abszolút címzés alatt egy meghatározott programtárolóhelyre (nnn) való ugratást értünk.

Pl. készítsünk programot a 4-esével való számoláshoz!

Billentyű	Megjegyzés
2nd CP	a programtároló törlése, beállítás a 000 tárolóhelyre
LRN	a tanulási üzemmód beírása
2nd Lbl SUM	a programszakaszt a SUM jelzéssel látjuk el
+ 4 =	minden eredményhez 4-et hozzáadunk
2nd Pause	szünet és a szám kijelzése
GTO SUM	átmenet a SUM címkehez és a műveletek ismétlése
LRN	a tanulási üzemmód feloldása
GTO SUM	a programmutató beállítása a SUM címke.

Ha most megnyomjuk az R/S billentyűt, akkor a kijelzőben 4, 8, 12 stb. jelenik meg. A készülék a +4 = műveletsort ismételtelen végrehajtja, amíg az R/S billentyűvel le nem állítjuk.

Ez a program többféleképpen megírható. Alapfeltétel azonban, hogy minden műveletsor a 000 tárolóhellyel kezdődjék!

Billentyű	Billentyű	Billentyű	Billentyű
2nd Lbl	+	2nd Lbl	+
SUM	4	C	4
+	=	+	=
4	2nd Pause	4	2nd Pause
=	GTO	=	RST
2nd Pause	0	2nd Pause	
GTO	0	C	
SUM	0		
általános címke	abszolút címzés	programcímke	visszaállítás

Az általános címke módszere látható az első oszlopban. Ha abszolút címzéssel dolgozunk, főlegesek a címkek. Ha ez a szakasz a programtárolóban egy későbbi helyen áll és alacsonyabb sorszámu címre kell helyesbíteni, akkor bizonyos esetekben az abszolút címet is meg kell változtatni. A C billentyű segítségével a program végrehajtása mindig visszatér C programcímkehez, és a számítás automatikusan végbemegy. Ez a módszer lefoglalja az alprogram-visszaléptető tárolót is.

A háromjegyű abszolút cím tárolásához két tárolóhelyre van szükségünk. A GTO 126 utasításhoz pl. három hely szükséges: GTO 01 26. Ha 100 alatti címekkel dolgozunk, csak a cím jelentéssel bíró részét kell beírni. Ha GTO 7 utasítást billentyűzzünk, akkor ez a rövid alakú címzés automatikusan három tárolóhelyet foglal el: GTO 00 07. A GTO utasítást mindig ki kell egészíteni egy legalább egy-jegyű címmel, különben a következő beírást a készülék címkeként tekinti.

Alprogramok

SBR (subroutine) N vagy nnn: olyan utasítássorozat, amelyet a program főrészéről leválasztunk valamely matematikai vagy logikai művelet elvégzésére. Ezt az utasítást a főprogram vagy egy másik alprogram előhívhatja és végrehajthatja. Az alprogramok olyan helyzetekben használhatók, amikor a program egy vagy több különböző helyeken ismétlődő programlépéssorozatot tartalmaz. Ezt a lépéssort — annak alkalmankénti ismételt beírása helyett — csak egyszer kell beírni, de szükség szerinti alkalommal lehet előhívni. Miután az alprogramot előhívtuk, és a feladatát végrehajtotta, a vezérlés visszakérül ahhoz a programcímhez, amely közvetlenül az alprogramot behívó utasítás után áll.

Tanulási üzemmódban az SBR utasítással átmenetet hajtunk végre ahhoz az alprogramhoz, amelynek

címkeje N vagy tárolóhelye nnn. Ha az alprogramot végrehajtás közben hívjuk elő, a feldolgozási folyamat azonnal az alprogramra irányul. Az alprogram behívása után a tárolóhely száma az alprogram-visszaléptető-tárolóban van. Minden alprogram INV SBR utasítással végződik. A visszalépés arra a tárolóhelyre történik, amely az alprogram-visszaléptető tárolóban van; a feldolgozás innen folytatódik. Az INV SBR utasítás az R/S utasításhoz hasonlóan működik, ha azt nem alprogramban alkalmazzuk. Az alprogram-visszaléptető tároló kapacitása 6 visszaléptetési cím. Ily módon lehetséges, hogy egy alprogram behívhat egy másikat. Ha egy alprogramot lezárunk, a hozzárendelt visszaugrási cím törlik a tárolóból, hogy szükség esetén helyet adjon egy új alprogramnak.

Ha az alprogram-utasítást billentyűzve adjuk be, a feldolgozás automatikusan folytatódik a kívánt címkénél vagy címnél!

Pl. számítsuk ki az $x^2 - 3x - 2$ kifejezést x_1 , és x_2 értéke mellett! Tároljuk az x_1 értékét az R01-ben, az x_2 értékét az R02-ben és a két eredményt összegezzük az R03-ban!

Mint ahogy az $x^2 - 3x - 2$ kifejezést kétszer kell kiszámítani, ezért erre a feladatra alprogramot készítünk. A 030 címnél való kezdés önkényes. Nyomjuk meg GTO 30 LRN billentyűt és írjuk be a következő utasítást!

Tárolás és billentyűkód	Billentyű	Megjegyzés
030 76	2nd Lbl	a CE címkével jelzett programszakasz
031 24	CE	
032 53	(	
033 42	STO	az x értékének tárolása a későbbi alkalmazáshoz
034 05	5	
035 33	x^2	
036 75	—	
037 03	3	
038 65	X	
039 43	RCL	az x értékének előhívása
040 05	5	
041 75	—	
042 02	2	
043 54	)	a kifejezés számítása
044 44	SUM	
045 03	3	
046 92	INV SBR	visszatérés a főprogramhoz.

Figyeljük meg, hogy a kifejezés számításához zárójeleket alkalmazunk! Egyenlőségi utasítást is alkalmazhattunk volna, ez azonban a fő- és alprogram összes befejezetlen műveleteit lezárja. A zárójelek használatának kockázatmentes. Ezután LRN RST LRN billentyűzése után megírhatjuk a főprogramot, majd az x értékét behelyettesítve jutunk el az eredményhez.

Tárolóhely és billentyűkód	Billentyű	Megjegyzés
000 43	RCL	az x_1 előhívása
001 01	1	
002 71	SBR	a CE alprogram előhívása
003 24	CE	
004 43	RCL	az x_2 előhívása
005 02	2	
006 71	SBR	a CE alprogram előhívása
007 24	CE	
008 43	RCL	az eredmény előhívása a 03 tárolóba
009 03	3	
010 91	R/S	leállítás, az R03 tartalmának kijelzése
011 81	RST	visszaállítás 000-ra (a következő változóhoz), az alprogram-visszaléptető tároló visszaállítása 0-ra.

Ha az alprogramot előhívjuk, a feldolgozási folyamat a CE címkénél folytatódik (030 tárolóhely), az alprogram lezárul (a visszaléptetési cím a 004 tárolóhely). A következő alprogram visszaléptetési helye a 008 tárolóhely. A visszatérés az alprogramból mindig ahhoz a tárolóhelyhez történik, amely az alprogramot előhívó utasítás után áll.

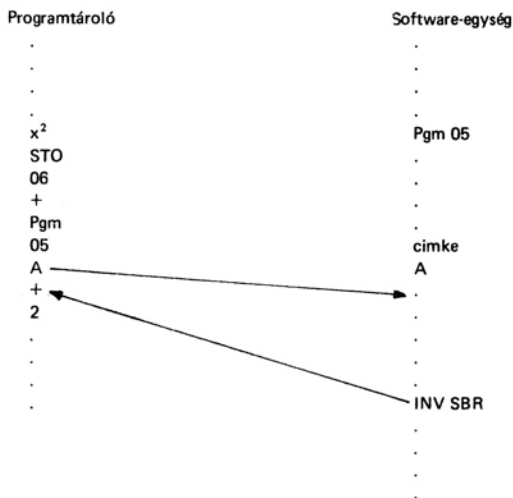
Ehhez még hozzá kell tennünk, hogy a főprogram által előhívható alprogramokat más alprogramok is előhívhatják. Az alprogram-visszaléptető tároló automatikusan ellenőrzi, hogy egy alprogram lezárása után hova kell a visszaléptetésnek történnie.

Ha egy program billentyűzése után benyomjuk az RST billentyűt, hogy visszatérjünk a 000 tárolóhelyre, ezzel egyidejűleg az alprogram-visszaléptető tárolót is nulláztuk.

Software-programok mint alprogramok

A készülékbe helyezett software-egység bármelyik programja előhívható alprogramként. A programokat különösen erre a célra alakítottuk ki, vagyis egy program minden része alprogram. Minden részt címkével jelöltünk, vagyis INV SBR utasítással végződik és többnyire nem tartalmaz R/S utasítást. Az egyes programrészekkel kapcsolatos tudnivalók a software-programokat tartalmazó füzetben találhatók.

Már tudjuk, hogyan férhetünk hozzá a software-programokhoz a billentyűzet útján. Ha a programtárolóban levő program befejeződik, előhívhatunk egy software-programot és ennek tetszőleges részeit alkalmazhatjuk alprogramként. Ha a szakaszt egy programcím-billentyűvel jelöltük, a 2nd Pgm mm N billentyűzés a program az mm software-programnál folytatódik, illetve a végrehajtás a software-programban az N címkénél történik. Egy software-programban egy általános címkéhez a következő billentyűzéssel jutunk el: 2nd Pgm mm SBR N. Ha az N címkével ellátott software-programszakasz teljesítette feladatát, a programszámláló arra az utasításra irányul, amely a programtárolótól való elágazási hely után áll, így a feldolgozás innen folytatódik.



Egy programban a 2nd Pgm mm N utasításhoz hasonló az SBR N utasításhoz. Az ezen utasítás utáni cím az alprogram-visszaléptető tárolóban van, így a feldolgozást azonnal az N címkével jelölt program-részre irányítjuk. Ha az N címke utáni utasítássort lezártuk, a feldolgozást annál a címnél folytatjuk,

amelyet utolsóként tároltunk az alprogram-visszaléptető tárolóban. A 2nd Pgm mm utasítás után SBR-utasításnak vagy egy programcím-billentyűnek kell következnie. Más billentyűzés nem engedhető meg, mert hibás eredményhez vezet.

A software-programoknak alprogramként való alkalmazása növeli a számológép kapacitását. Van azonban néhány megfontolandó szempont, ha két különálló programot vonunk össze. Programoknak sohasem szabad egymást átfedni, azonos adattárolók és jelzők nem lehetnek. Egy software-program más alprogramokat is előhívhat, azonban ügyelnünk kell arra, hogy legfeljebb 6 alprogramsíkot vehetünk igénybe. Általában nem lépünk túl ezt a kapacitást, ha legfeljebb 3 alprogramsíkot használunk a programtárolóban. Ehhez még hozzá kell tenni, hogy a software-programok kijelzési módja is beállítható rögzített tizedes-pont, hatványkitevős, vagy műszaki kijelzésre.

A következő táblázatban röviden összefoglaltuk az alprogramok különböző alkalmazási lehetőségeit.

Programtárolótól vagy software-programtól	software-programhoz	2nd Pgm mm, N (programcímke) 2nd Pgm mm címke SBR N (általános címke) vagy 2nd Pgm mm R/S SBR nnn vagy N
Programtárolótól	programtárolóhoz	2nd Pgm 00 N (vagy nnn) vagy
Software-programtól	programtárolóhoz	2nd Pgm 00 SBR N vagy RST

Feltételes átmenetek (vizsgáló utasítások)

Ezen utasítások hatására csak akkor ágazik el a program, ha a kijelzőtárolóban levő érték bizonyos feltételeknek eleget tesz. Ezek: a kijelzőtároló pillanatnyi értéke a t-tároló értékéhez képest, továbbá az előírt számú lépés feldolgozása megtörtént-e?

Összehasonlítás a t-tárolóval

$x \approx t$	a kijelzőtárolóban levő x érték és a t-tárolóban levő t érték felcserélése
2nd x = t N vagy nnn	egyenlő a kijelzőtároló tartalma a t-tároló tartalmával?
INV 2nd x = t N vagy nnn	nem egyenlő a kijelzőtároló tartalma a t-tároló tartalmával?
2nd x > t N vagy nnn	a kijelzőtároló tartalma nagyobb vagy egyenlő a t-tároló tartalmával?
INV 2nd x > t N vagy nnn	a kijelzőtároló tartalma kisebb a t-tároló tartalmával?

Ha a fenti kérdések valamelyikére igen a válasz, a feldolgozás arra a címre ágazik el, amely közvetlenül az utasítás után áll. Ha a válasz nem, akkor a megfelelő címet a program átugorja és a feldolgozás a következő utasítással folytatódik. Ezek a vizsgálati utasítások nincsenek hatással a befejezetlen műveletekre, és tetszőleges alkalmakkal végrehajthatók egy programon belül.

Tárolóhely és billentyűkód	Billentyű
022 43	RCL
023 01	1
024 95	=
025 77	2nd x > t
026 12	B
027 61	GTO
028 00	3
029 03	
030 76	2nd Lb1
031 12	B
032 91	R/S

A 024 helyen tárolt számot a 025 helyen hasonlitsuk össze a t-tároló tartalmával, hogy az nagyobb vagy egyenlő-e az utóbbival! Ha a válasz igen, a feldolgozás annál a címkénél folytatódik, ahol a program megszakad, és az eredményt a készülék kijelzi. Nemleges válasz esetén a B címkéhez vezetődő elágazási utasítást átugorjuk. A GTO 003 átmeneti utasítással a feldolgozás a 003 címnél folytatódik.

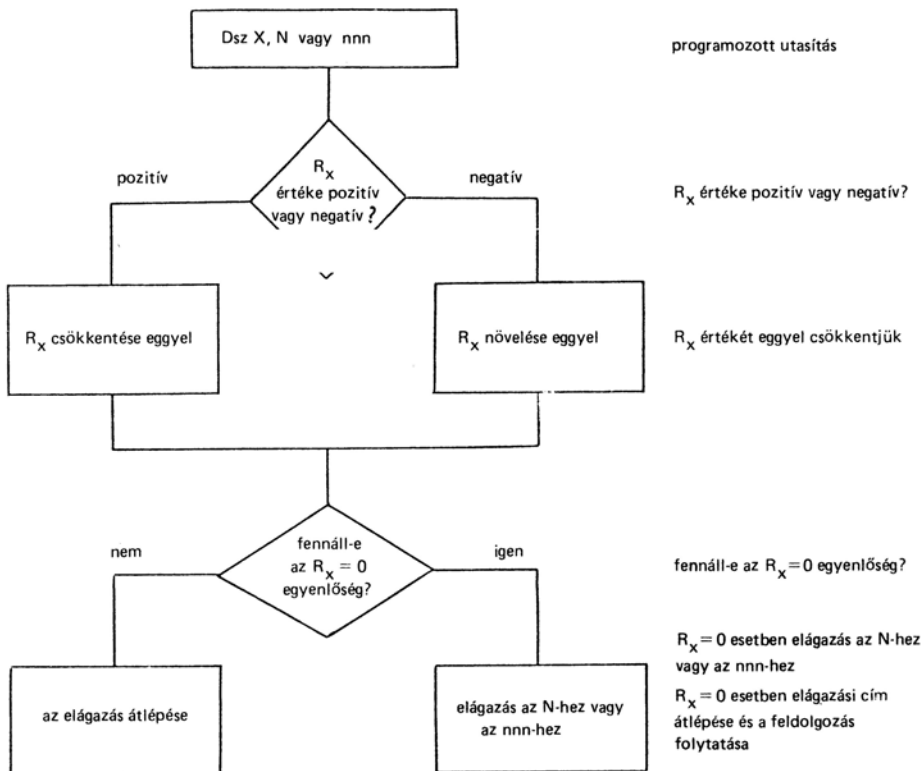
Csökkentés és átmenet 0-nál

Csökkentés és átmenet 0-nál: 2nd Dsz (decrement and skip on zero) X, N vagy nnn.

Az adattároló tartalmát 0...9-cel csökkentjük; ha $R_x \neq 0$, a végrehajtás az N címkénél vagy az nnn tárolóhelyénél folytatódik. Az $R_x = 0$ esetben viszont az elágazási címét átlépjük. (R_x az X adattároló tartalmát jelenti).

Megjegyzés: a feltételes elágazásnál a visszaléptetési címek nem az alprogram-visszaléptető tárolóban vannak. Ha elő kell hívunk egy alprogramot, és vissza kell térnünk egy feltételes elágazásra, akkor a vizsgálat az alprogram első lépése legyen! 2nd $x = t \cdot x^2$ utasítás pl. nem tartalmaz visszaléptetési címet. Az SBR x^2 és a 2nd Lb1 x^2 2nd $x = t$ utasítás viszont tárolja a visszaléptetési címet.

Ez egy összevont programutasítás. Ha egy utasítást y-alkalommal kell megismételni, az y értékét a 0...9 adattárolók egyikében tároljuk, és Dsz utasítást programozunk ismétlődő sorrendben. Az y-ismétlődése után a hurok befejeződik, és a program folytatódik. A Dsz utasítás működése:



Az ábra mutatja, hogy a Dsz utasítással kapcsolatos döntés megelőzi a számítást. Az utasítássor végén a Dsz utasítás szerint először függvényt számít a gép, csak azután számlál tovább. Egy program utasításainak számát helyesen úgy kapjuk, hogy beírjuk az y -értéket az X-tárolóba majd végrehajtjuk a Dsz utasítást. A másik megoldás szerint először végrehajtjuk a Dsz utasítást, majd beírjuk az $y + 1$ értékét.

Pl.: írjunk programot az $F!$ (F faktoriális) számítására! $F! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot F$. A megállapodás szerint $0! = 1$.

Tárolóhely és billentyűkód	Billentyű	Megjegyzés
000 76	2nd LbI	
001 15	E	
002 42	STO	az F értéke a 00 tárolóban
003 00	0	
004 29	2nd CP	a t-tárolás törlése
005 67	2nd $x = t$	vizsgálat $F = 0?$
006 11	A	ha igen, átmenet A-hoz
007 76	2nd LbI	
008 12	B	
009 43	RCL	az F előhívása
010 00	0	
011 65	X	
012 97	2nd Dsz	az R_{00} csökkentése eggyel
013 00	0	
014 12	B	ha az R_{00} nincs a ± 1 tartományban, átmenet a B-hez
		ha az R_{00} a ± 1 tartományban van, a program folytatódik
015 76	2nd LbI	
016 11	A	
017 01	1	
018 95	=	
019 91	R/S	megállás és az $F!$ kijelzés

A műveletsor végrehajtására adjunk be egy F értéket, és nyomjuk meg E billentyűt! Az F értékének 70-nél kisebbnek kell lenni, különben a gép kapacitását túllépjük; ugyanis $70! > 9,9999999 \cdot 10^{99}$

Jelzők

2nd St flg (set flag) y . Ez az utasítás beiktat egy y jelzőt, ahol $y = 0 \dots 9$. Az INV 2nd St flg utasítással az y jelzőt kiiktatjuk.

2nd If flg y , N vagy nnn. Az y jelzőt megvizsgáljuk, hogy beiktattuk-e. Ha igen, átmenet történik az N-címkehez vagy az nnn tárolóhelyhez.

INV 2nd If flg y ; N vagy nnn. Az y jelzőt megvizsgáljuk; ha az y jelzőt nem iktattuk be, a program elágazik az N címhez vagy az nnn tárolóhelyhez.

A programban való alkalmazásra 10 jelző (0...9) szolgál. Ezek a kezdetben 0-ra vannak állítva. A jelzőket vagy a számítási útvonal ellenőrzésére használjuk a program végrehajtása közben, vagy a programelöváltásztás kézi vezérlésére a végrehajtás előtt; ha RST vagy 2nd CP utasításokat billentyűzünk, az összes jelzőt nullázzuk, s egyben töröljük az alprogram-visszaléptető tárolót. A programban lévő RST utasítás szintén nullázza a jelzőket.

Pl.: készítsünk olyan programot, amely összegzi a beírt számokat, és a pozitív számokat kinyomtatja! Az összeget minden beírás után ki kell jelezni.

Tárolóhely és billentyűkód	Billentyű	Megjegyzés
018 76	2nd Lbl	
019 11	A	
020 22	INV	
021 86	2nd St flg	a 3. jelzőt 0-ra kell állítani
022 03	3	
023 77	2nd $x \geq t$	a szám 0 vagy pozitív?
024 12	B	ha igen, átmenet a B címkehez
025 86	2nd St flg	negatív számnál beiktatjuk a 3. jelzőt
026 03	3	
027 76	2nd Lbl	
028 12	B	
029 44	SUM	a beírt számok összegezése
030 12	1 2	
031 87	2nd If flg	léptetve van a 3. jelző?
032 03	3	
033 13	C	ha igen (neg. szám) átmenet a C címkehez
034 99	2nd Prt	ha nem, a szám kinyomtatása
035 76	2nd Lbl	
036 13	C	
037 43	RCL	
038 12	1 2	
039 91	R/S	

Ellenőrizzük a számsor beírása előtt, hogy a 12. adattároló nem foglalt-e! Ha a számot beírjuk, és az A billentyűt megnyomjuk, akkor a számot hozzáadjuk a 12 tároló tartalmához. A pozitív beírást kinyomtatja, a beírt számok összegét pedig kijelzi a gép.

Jelzők és hibafeltételek

A 8. jelző arra szolgál, hogy egy programművelet során a program hibaállapotát meghatározzuk vele. Általában a programvégrehajtás hibafeltétel esetén is folytatódik. Teljesen mindegy, hogy a 8. jelzőt kézzel billentyűzzük vagy a programban előre iktatjuk be; mindkét esetben megszakad a program, ha a hibafeltétel teljesül.

2nd Op 18 utasítással léptetjük a 7. jelzőt, ha a programban nincs hibafeltétel.

2nd Op 19 utasítással léptetjük a 7. jelzőt, ha a programban hibafeltétel van. Így a 7. jelző a program hibafeltételeinek ellenőrzésével módot ad a szükséges beavatkozásra. Ha a vizsgálatok valamelyike hibás, akkor a 7. jelző nem változik.

KÖZVETETT CÍMZÉS

2nd Ind XX. Ha ezt az utasítást a következő műveletek valamelyike után írjuk be, akkor előhívja az XX tárolót. Ennek tartalma a helyes elágazási cím vagy olyan adattároló címe, amelyet a további végrehajtásnál alkalmazni kell.

Ügyeljünk arra, hogy a közvetett tárolási műveleteknél a Pgm, az Op és az átmeneti utasításhoz olyan kódok tartoznak, amelyek nincsenek összefüggésben az említett billentyűk helyzetével. Ezen kódok hozzárendelése a programtárolóhelyekkel való takarékoskodás alapján történt.

Billentyű	Billentyűkód	Megjegyzés
STO 2nd Ind XX	72 XX	közvetett tárolás
RCL 2nd Ind XX	73 XX	közvetett előhívás
2nd Exc 2nd Ind XX	63 XX	közvetett csere
SUM 2nd Ind XX	74 XX	közvetett tároló-összeadás
INV SUM 2nd Ind XX	22 74 XX	közvetett tároló-kivonás
2nd Prd 2nd Ind XX	64 XX	közvetett tároló-szorzás
INV 2nd Prd 2nd Ind XX	22 64 XX	közvetett tároló-osztás
GTO 2nd Ind XX	83 XX	közvetett átemeneti utasítás
2nd Pgm 2nd Ind XX	62 XX	közvetett programelőhívás
2nd Op 2nd Ind XX	84 XX	közvetett vezérlési művelet
SBR 2nd Ind XX	71 40 XX	közvetett alprogram
2nd $x \geq t$ 2nd Ind XX	77 40 XX	közvetett $x \geq t$ vizsgálat
INV 2nd $x \geq t$ 2nd Ind XX	22 77 40 XX	közvetett $x < t$ vizsgálat
2nd $x = t$ 2nd Ind XX	67 40 XX	közvetett $x = t$ vizsgálat
INV 2nd $x = t$ 2nd Ind XX	22 77 40 XX	közvetett $x \neq t$ vizsgálat
2nd Fix 2nd Ind XX	58 40 XX	közvetett rögzített tizedespont-beállítás
2nd St flg 2nd Ind XX	86 40 XX	közvetett jelzőléptetés
INV 2nd St flg 2nd Ind XX	22 86 40 XX	közvetett jelző-visszaléptetés
2nd Dsz 2nd Ind XX,		
N vagy nnn	97 40 XX	közvetett tároló Dsz vizsgálat
2nd Dsz X 2nd Ind XX	97x40 XX	közvetett cím Dsz vizsgálat
2nd Dsz 2nd Ind XX		
2nd Ind yy	97 40 XX 40 yy	közvetett tároló és cím Dsz vizsgálat
INV 2nd Dsz 2nd Ind XX,		
N vagy nnn	22 97 40 XX	közvetett tároló átmenet a nem 0 eredmény esetén
INV 2nd Dsz X 2nd Ind XX	22 97 X 40 XX	közvetett cím-átmenet nem 0 eredmény esetén
INV 2nd Dsz 2nd		
Ind XX 2nd Ind yy	22 97 40 XX 40 yy	közvetett cím és tároló-átmenet nem 0 eredmény esetén
2nd If flg 2nd Ind XX,		
N vagy nnn	87 40 XX	közvetett jelzőszám, jelzővizsgálat
2nd If flg X 2nd Ind yy	87x40 yy	közvetett cím, jelzővizsgálat
2nd If flg,		
2nd Ind XX 2nd Ind yy	87 40 XX 40 yy	közvetett jelzőszám és cím, jelző vizsgálat
INV 2nd If flg 2nd Ind XX,		
N vagy nnn	22 87 40 XX	közvetett jelzőszám, a jelzővizsgálat megfordítása
INV 2nd If flg 2nd		
X 2nd Ind XX	22 87 X 40 XX	közvetett cím, a jelzővizsgálat megfordítása
INV 2nd If flg 2nd Ind XX		
2nd Ind yy	22 87 40 XX 40 yy	közvetett jelzőszám és cím, a jelzővizsgálat megfordítása

A közvetett utasítás a címet olyan alakúra módosítja, hogy az az adattárolóban megtalálható legyen. Tehát a RCL 2nd Ind 04 utasítással nem 04 tároló tartalmát hívjuk elő, hanem az R₀₄ adja meg a kívánt tároló-címet. Ha a 04 tárolóban a 111 szám van, akkor 111 lesz a kívánt cím. A közvetett cím a tényleges címet mutatja, amit a következő ábra szemléltet.

A KA-100 sornyomatónak külön használati utasítása van, de itt is közlünk néhány fontos tudnivalót.

Nyomtatás szükség szerint

A 2nd Prt utasítás billentyűzésekor (vagy ha az egy programban előfordul), akkor a gép kijelzett értéket kinyomtatja. Teljesen mindegy, hogy a nyomtató Prt vagy a számológép 2nd Prt gombját nyomjuk meg.

Figyeljük pl. meg a 4 egészszámú többszöröseit számító programot!

Tárolóhely és billentyűkód	Billentyű
000 85	+
001 04	4
002 95	=
003 99	2nd Prt
004 81	RST

Adjunk be egy kezdő értéket, és nyomjuk meg az RST R/S gombot! Ezzel a következő nyomtatást kapjuk:

4.
8.
12.
16.
20.
24.
28.
32.
36.
40.
44.
48.
52.
56.
60.
64.
68.
72.
76.
80.
84.
88.
92.
96.
100.

A software-programok egysége egész csoport nyomtatási utasítást tartalmaz. Ha tehát rendelkezésünkre áll a KA-100 készülék, akkor a software-programok adatbeírásait és eredményeit automatikusan kinyomtatathatjuk.

A kinyomtatott eredmények csoportjait egymástól a 2nd Adv papírtovábbító billentyűvel választhatjuk el. Az utasítás billentyűzésekor (vagy valamely programban előfordulásakor), a papír nyomtatás nélkül továbblép. Tekintsük pl. a 4. többszöröseit számító programot! A kívánt tárolóhelyen az RST utasítás előtt iktassuk be a 2nd Adv utasítást, mire az alábbi ábra szerinti nyomtatást kapjuk. A papír elöretolásának természetesen nincs befolyása a számításokra.

- 4.
- 8.
- 12.
- 16.
- 20.
- 24.
- 28.
- 32.
- 36.
- 40.
- 44.
- 48.
- 52.

Több üres sort úgy kapunk, ha az utasítást többször megismételjük. A sornyomtatón levő ADV ↑ billentyű szintén üres sorokat hoz létre. A papír addig szalad előre, míg a billentyűt benyomva tartjuk.

Saját programok feljegyzése

Egy program feljegyzése céljából nyomjuk meg a 2nd List billentyűt! A gép a programot attól a helytől kezdve jegyzi, ahol a programszámláló éppen áll. A felsorolást az R/S billentyű megnyomásával bármikor megszakíthatjuk. Egy teljes program felsorolására az RST 2nd List billentyűzés szükséges. Az alábbi ábra mutatja a 4 többszöröseinek számítására szolgáló programot.

```

000 85 +
001 04 4
002 95 =
003 99 PRT
004 98 ADV
005 81 RST
006 00 0
007 00 0

```

Az adattárolók felsorolása

Az INV 2nd List utasítás billentyűzésekor (vagy programban előfordulásakor) a nyomtató felsorolja az adattárolók tartalmát; kezdve azzal a tárolócímmel, amely a kijelzőben látható. A felsorolás addig folytatódik, amíg a legmagasabb című adattároló tartalma is felsorolásra került, vagy amíg az R/S billentyűvel a nyomtatást meg nem szakítjuk. Ezzel a számológépet ismét billentyűvezérlésre kapcsoljuk. Az 50..59 tárolókra vonatkozó, tetszés szerinti tárolótartalmak felsorolása látható a következő ábrán:

	0.	47
	0.	48
	0.	49
14.	18181818	50
665.	8568182	51
110.	9761364	52
	0.	53
	0.	54
-5.	5488068-12	55
	0.	56
	0.	57
	1.	58
0.	085106383	59

A felsorolásra utasító billentyűzés 50 INV 2nd List.

Ha ezt a műveletsort egy programba beiktatjuk, akkor a számítások leállnak. A kívánt adattároló-tartalmaikat a gép felsorolja, majd a program megszakad, és a vezérlés visszakérül a billentyűzethez.

A számítás jegyzőkönyvezése, párhuzamos üzem

A nyomtatón levő TRACE billentyű benyomásával egy számítás minden egyes lépése kinyomtatható.

A készülék a számított értéket és a megfelelő utasítást kijelzi. Ez a kézzel billentyűzött és a programozott számításokra egyaránt érvényes.

A TRACE billentyű egy öntartó kapcsoló, csak újabb megnyomásra oldódik ki.

A 4 többszöröseit számító programhoz nyomjuk meg a nyomtatón levő TRACE billentyűt, majd a számológépen levő CLR RST R/S billentyűt! Így az ábra szerinti jegyzőkönyvet kapjuk a számításokról.

	CLR
	RST
0.	+
4.	=
4.	
4.	PRT
	RST
4.	+
4.	=
8.	
8.	PRT

	RST
8.	+
4.	=
12.	
12.	PRT

A programot és a jegyzőkönyvezést az R/S utasítással állíthatjuk le.

Ellenőrző jelölések párhuzamos üzemmódban

A kinyomtatott jelölések többsége nehézségek nélkül azonosítható, míg mások hozzárendelése az első pillantásra kétségesnek tűnik. A vezérlő szimbólumok teljes felsorolása a hozzájuk tartozó billentyűzéssel együtt:

Nyomtatási		Nyomtatási		Nyomtatási	
jelölés	Billentyűzés	jelölés	Billentyűzés	jelölés	Billentyűzés
A – E	A – E	ILOG	INV 2nd log †	RCL	RCL
A' – E'	2nd A' – 2nd E'	IND	2nd Ind	R/S	R/S
ADV	2nd Adv	INS	lásd a megjegyzés	RST	RST
BST	lásd a megjegyzést	INT	2nd Int	RTN	INV SBR
CE	CE	INV	INV	SBR	SBR
CLR	CLR	IPD*	INV 2nd Prd 2nd Ind†	SIN	2nd sin
CP	2nd CP	IP/R	INV 2nd P→R †	SM*	SUM 2nd Ind
CMS	2nd CMs	IPRD	INV 2nd Prd †	SST	lásd a megjegyzést
COS	2nd cos	ISBR	INV SBR †	ST*	STO 2nd Ind
DEG	2nd Deg	ISIN	INV 2nd sin †	STF	2nd St flg
DEL	lásd a megjegyzést	ISM*	INV SUM 2nd Ind †	STO	STO
DMS	2nd D.Ms	ISTF	INV 2nd St flg †	SUM	SUM
DSZ	2nd Dsz	ISUM	INV SUM †	TAN	2nd tan
EE	EE	ITAN	INV 2nd tan †	WRT	2nd Write
ENG	2nd Eng	Ix	INV 2nd x †	X ≅ T	x ≅ t
EQ	2nd x = t	IXI	2nd x	x ²	x ²
EX*	2nd Exc 2nd Ind	IY ^x	INV y ^x †	x̄	2nd x̄
EXC	2nd Exc	LBL	2nd Lbl	IXI	2nd x
FIX	2nd Fix	LNX	Inx	1/X	1/x
GE	2nd x ≥ t	LOG	2nd log	√x	√x
GO*	GTO 2nd Ind	LRN	lásd a megjegyzést	Y ^x	y ^x
GRD	2nd Grad	LST	2nd List		jelölések
GTO	GTO	NOP	2nd Nop		
IEQ	INV 2nd x = t †	OP	2nd Op		Σ+
IGE	INV 2nd x ≥ t †	OP*	2nd Op 2nd Ind		π
I Σ+	INV 2nd Σ+ †	PAU	2nd Pause		)
ICOS	INV 2nd cos †	PD*	2nd Prd 2nd Ind		(
IDMS	INV 2nd D.Ms †	PG*	2nd Pgm 2nd Ind		–
IDSZ	INV 2nd Dsz †	PGM	2nd Pgm		+
IFF	2nd If flg	P/R	2nd P→R		x
IFIX	INV 2nd Fix †	PRD	2nd Prd		÷
IIFX	INV 2nd If flg †	PRT	2nd Prt		=
IINT	INV 2nd Int †	RAD	2nd Rad		.
ILNX	INV Inx †	RC*	RCL 2nd Ind		+/-

Megjegyzés: A megjelölt utasítások csak akkor láthatók, ha a billentyű a programjegyzékben előfordul. Mivel a billentyűkód nem vihető be a programtárolóba, ha megnyomjuk ezt a billentyűt, a billentyűkód csak egy másik utasítás átalakítása utáni maradvány lehet, így javítani kell.

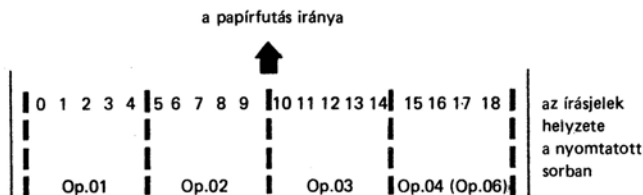
A † jelű utasítás csak jegyzőkönyvezésnél használható.

Különleges vezérlési műveletek

A 00..08 vezérlési műveletek a nyomtatóval való együttműködés céljára szolgálnak.

Írásjegyek nyomtatása – Op 00..06

Az első hét vezérlési művelettel írásjegyeket (számokat és betűket) nyomtathatunk ki. Minden sorba 20 írásjel kerülhet. Ezeket ötösével csoportosítjuk és tároljuk, amint ez az ábrán is látható.



Minden kinyomtatott írásjegyet egy kétjegyű kód jellemez. A kódot a sor-oszlop címből származtatjuk.

		EGYESEK							
		0	1	2	3	4	5	6	7
TÍZESEK	0		0	1	2	3	4	5	6
	1	7	8	9	A	B	C	D	E
	2	–	F	G	H	I	J	K	L
	3	M	N	O	P	Q	R	S	T
	4	.	U	V	W	X	Y	Z	+
	5	X	*	√	π	e	(	)	,
	6	↑	%	≥	/	=	'	x	$\bar{x}$
	7	²	?	÷	!		△	Π	Σ

Pl. az A kódja a 13, a + kódja a 47. Öt írásjel kódja, azaz tíz hely egyidejűleg beírható a kijelzőbe. Ha nem töltjük ki a 10 lehetséges helyet, akkor a beírt értékek elé automatikusan nullák kerülnek. Két nulla az üres hely kódja; az üres helyek kitöltésére két nullát kell beírni.

Amennyiben a kijelző írásjegy-kódot tartalmaz, nyomjuk meg a 2nd Op 01, 02, 03 vagy 04 billentyűt, ezzel a számológépet arra utasítjuk, hogy a sor melyik helyére nyomtassa ki az írásjeleket!

2nd Op 01 a sor bal külső negyedére

2nd Op 02 a sor bal belső negyedére

2nd Op 03 a sor jobb belső negyedére

2nd Op 04 a sor jobb külső negyedére

A 2nd Op 00 utasítással töröljük a nyomtatótárolót. A 2nd Op 05 utasítással a nyomtatótárolót tartalmát nyomtatjuk ki. A felirat felhasználható a papírtekercs részeinek megjelölésére, de felhasználható egy programban is.

Példa: írjuk papírszalagra a π^2 VS X% TESTS 3/22'' feliratot!

π	²	V	S	X	%	T	E	S	T	S	S	3	/	2	2	
53	70	00	42	36	00	44	61	00	37	17	36	37	36	00	04	63

Billentyű	Kijelzés	Megjegyzés
CLR 2nd Op 00	0.	a nyomtatóregiszter törlése
5370004236 2nd Op 01	5370004236.	a π^2 VS tárolása, nyomtatás a bal külső negyedben
44610037 2nd Op 02	44610037.	az X% T tárolása, nyomtatás a bal belső negyedben
1736373600 2nd Op 03	1736373600.	az ESTS tárolása, nyomtatás a jobb belső negyedben
463030300 2nd Op 04	463030300.	a 3/22 tárolása, nyomtatás a jobb külső negyedben
2nd Op 05	463030300.	a teljes felirat kinyomtatása.

Ügyeljünk arra, hogy a nyomtatott sor belső negyedében egy üres helyre van szükség! A nyomtató tároló be nem töltött nullái hozzák létre ezt az üres helyet. A jobb külső negyedben 04 az első szükséges írásjel, de csak 4-et kell beírni. A sornegyedeket tetszőleges sorrendben írhatjuk meg, azok más felírat nyomására is utasíthatók. Azok a tárolók, amelyeket a feliratokhoz lefoglaltunk, egyébként az 5., a 6., a 7. és a 8. befejezetlen művelethez szolgálnak tartalékul. Egy felírat összeállításakor tehát 4-nél több műveletet nem hagyhatunk befejezetlenül.

Ügyeljünk arra, hogy az írásjegyek beírásakor a kijelzés másodlagos formái (rögzített tizedespontos, hatványkifejtős és a műszaki kijelzési mód) megszűnnek.

További különleges vezérlési művelet a 2nd Op 06. Ennek segítségével a kijelző pillanatnyi értéket és a sor jobb külső negyedében levő 4 írásjegyet nyomtathatjuk ki. Ez módot ad a programeredmények megjelölésére.

Pl. készítsunk programot a π közelítő értékének, a 22/7-nek kiszámítására és megjelölésére!

Tárolóhely és billentyűkód	Billentyű	Megjegyzés
000 03	3	} P } I
001 03	3	
002 02	2	
003 04	4	
004 69	2nd Op	a „PI” tárolása a nyomtatáshoz
005 04	0 4	
006 02	2	
007 02	2	
008 55	÷	
009 07	7	
010 95	=	22 ÷ 7
011 69	2nd Op	nyomtatás
012 06	6	
013 91	R/S	

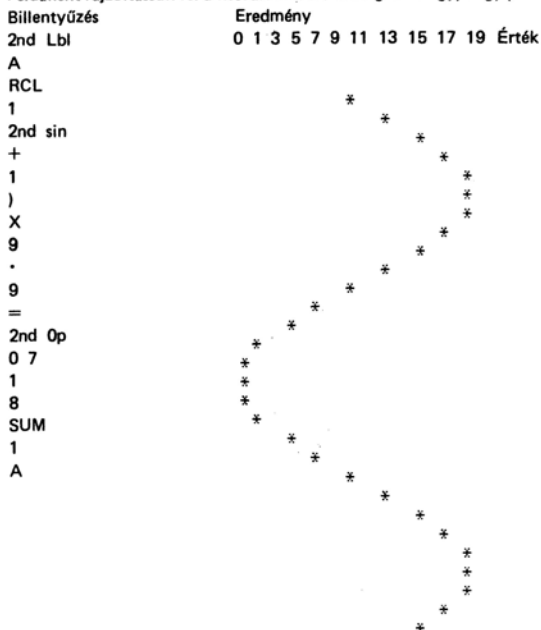
Ha a program végrehajtódik, a sornyomtató kinyomtatja a „3,142857143 PI” szöveget.

Az adatok ábrázolása Op 07

A 07 vezérlési művelet a 0...19. oszlopban pillanatnyilag kijelzett értéket balról jobbra növekvő értékrendben felrajzolja. A művelet — amelynek alkalmazása főként programon belül célszerű — lehetővé teszi görbék felrajzolását. Az X érték egész részének a 0...20 tartományba kell esnie. Ha a kijelzett érték nincs a tartományon belül, akkor azt a készülék nem nyomtatja ki, a programfutás megszakad és a kijelző villog.

A nyomtató csak az egész részt nyomtatja ki.

Példaként rajzoltassuk fel a készülékkel a szinuszgörbét úgy, hogy pontjait 18 fokként képezzük.



Ügyeljünk arra, hogy a programban kiszámított szinuszártékhoz 1-et hozzáad a készülék, hogy minden eredmény pozitív legyen. Így az értékek a 0...2 tartományba esnek. Ha a léptéket 9,9-szeresére növeljük, a tartomány 0..19,8 értékekre tevődik át. A program úgy megy végbe, hogy először a 01 tárolóban a kiindulási szöveget tároljuk, majd megnyomjuk az A billentyűt. A program végrehajtása addig tart, amíg meg nem szakítjuk az R/S billentyűvel.

A programban szereplő címkék felsorolása Op 08.

A 2nd Op 08 utasítás segítségével megkapjuk mindazon címkék és tárolóhelyek folyamatos felsorolását, amelyek a programtárolót pillanatnyilag elfoglalják. A felsorolás a program pillanatnyi helyzetéből indul ki. Tehát a címkék felsorolása érdekében a 2nd Op 08 előtt nyomjuk meg a GTO vagy az RST billentyűt!

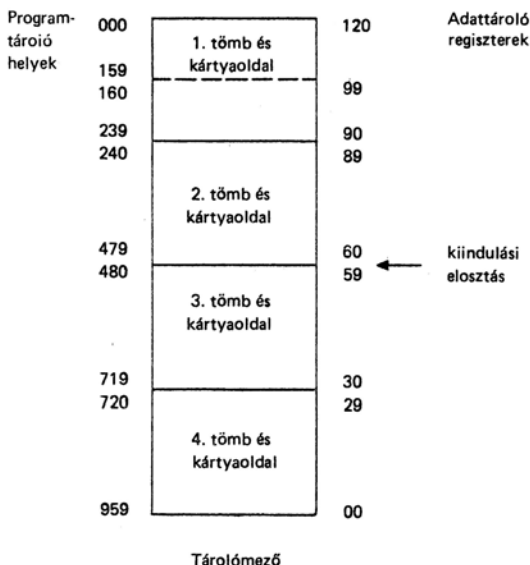
001	11	A
018	12	B
062	19	D'
129	13	C
205	88	DMS
239	70	RAD
273	80	GRD
282	14	D
288	15	E
294	16	A'
300	10	E'
306	28	LOG

A nyomtatófej tisztítása

A nyomtatófej tisztítására szolgáló eljárást KA-100 kezelési utasítás részletesen tartalmazza.

VII. MÁGNESKÁRTYÁK

A készülék tartozékként szállított mágneskártyákra minden tárolt programot vagy tárolt adatot tartósan rögzíthetünk. Amint azt már tudjuk, a készülék 120 tároló regiszterrel rendelkezik; ezek programtárolók, ill. adattárolók, egyenként 30 tárolót tartalmazó 4 tömbre vannak felosztva. A mágneskártya egy oldalára egy ilyen tömböt vagyis a kártya két oldalára két tömböt tudunk rögzíteni.



Rögzítés a mágneskártyára

A mágneskártyára a 2nd Write billentyűvel adott utasítással rögzíthetünk. Az n (1, 2, 3 vagy 4) tömb tartalmának rögzítéséhez nyomjuk meg az n 2nd Write billentyűt, és a kártyát nyomtatott oldalával felfelé toljuk be a készülék jobboldali alsó rekeszébe! Ügyeljünk arra, hogy a mágneskártyára történő átírásnál a rögzítéspontos kijelzési mód (kivéve a 2nd Fix 0 esetet) nem megengedett! Ha a kijelzési alakkal kapcsolatban bizonytalanok vagyunk, rögzítés előtt billentyűzzünk INV 2nd Fix utasítást! A kártya behelyezése az alábbi ábra szerint történik.



A tárolótömb tartalmát átvihetjük mágneskártyára, és teljesen mindegy, hogy a tömb programlépést, adatot, vagy mindkettőt tartalmaz. Az a kártyaoldal, amelyre rögzítünk, azt a tömbszámot viseli, amelyben az adatokat tároljuk. A 2nd Write billentyű működtetésekor a készülék a kijelzett számnak csak az egész részét veszi figyelembe, a törtrészt nem.

Például a 2.31 2nd Write utasítás a 2. tömb tartalmát a 2. jelű kártyaoldalra viszi át. Minden olyan értékre, amely 1-nél kisebb vagy 4-nél nagyobb, a kijelző villog, és nem történik mágneses rögzítés. Ügyeljünk tehát arra, hogy mágneses rögzítés esetén a kijelzett szám 1, 2, 3 vagy 4 legyen!

Amikor a készülékbe betolunk egy kártyát, ne akadályozzuk a mozgásban amikor már a szerkezet rögzítette! A mágneses rögzítés lezárásáig a kijelzés sötét marad, majd a készülék kijelzi a rögzített tömb számát. Ha rögzítés után a tömb száma villog, töröljük a kijelzést, és ismételjük meg a rögzítést! Ha a kijelzés még mindig villog, akkor valószínűleg hibás a mágneskártya. Ekkor kísérletezzünk egy másik kártyával!

Ha a program helyett adatokat akarunk rögzíteni, gondoljuk meg, hogy a 00 tároló a 4. tömb végén van, és az adattárolók számozása visszafelé a 3. tömbben folytatódik.

A mágneskártyát mindig a rajta tárolt információknak megfelelően feliratozzuk! A kártya két felső sarkában van hely a tömbszámok megadására. E mezőkben levő nyílak mutatják, hogy milyen irányban kell a kártyát a készülékbe tolni ahhoz, hogy a felírt tömbszámoknak megfelelő rögzítést kapjunk. A kártya középső, felső mezőjébe a programcímet és más jellemző adatot, pl. a tárolómező elosztását rögzíthetjük. Ez alatt két, részekre osztott sort találunk. Az alsó 5 négyzetben rögzítjük a programban előforduló A...E programcím-billentyűk rendeltetését. Ugyanígy használható a felettük levő sor a 2nd A...2nd E' billentyűk jelölésére. Egy tárolótömb rögzítésekor ügyelni kell arra, hogy a tömbszámot és a pillanatnyi tárolómező elosztást is rögzítsük a kártyán!

Példa a kártya felírti oldalának kitöltésére.

Kijelzés a 2nd Write billentyűzése és a kártya betolása után	Általános program	Védett program
1, 2, 3, 4	A tömb információjának beírása az egyik kártyaoldalra, (programlépés vagy adat) beleértve a tömbszámot és a pillanatnyi tárolóelosztást	Ha a tömb csak programlépéseket tartalmaz, nem történik rögzítés, és a szám a kijelzőben villog. Ha a tömb adatokat is tartalmaz, negatív tömbszámmal rögzítődik, és nem lesz védett.
-1, -2, -3, -4,	A kártyaoldalra beíródik és védett lesz a megadott tömbszámú, és a pillanatnyi tárolóelosztásnak megfelelő tömb információja	Ha a tömb csak programlépéseket tartalmaz, nem történik rögzítés, és a kijelzett szám villog. Ha a tömb adatokat is tartalmaz, negatív tömbszám rögzítődik
Más tetszőleges szám	A kártya átfut, de rögzítés nem történik. A kijelző jobb sarkában két számjegy villog	Mint az általános programnál

Ha mágneses rögzítés közben a kijelzőben egy szám villog, a kártya áthalad ugyan, de rögzítés nem történik.

A számológép kijelzőjét mágneses rögzítés esetén nem szabad rögzített tizedesponos alakra állítani. A kijelző villogása esetén a továbbító szerkezet tovább működik, amíg a kártya ki nem jön.

A programok adatvédelme

Ha mágneskártyára rögzített program titkos információkat is tartalmaz, akkor ezek védeltségét biztosítani lehet. Ez esetben a mágneses rögzítés előtt írjunk be negatív tömbszámot! Az $n \pm$ 2nd Write billentyűzés rögzíti a kártyán azt, hogy a programot újra beolvassuk a tárolóba, és újra végrehajthassuk. Ha a programot ismét beolvastuk a készülékbe, annak alkalmazásakor a következő korlátozások érvényesek:

1. nem történhet jegyzőkönyvezés és felsorolás sornymotatóval,
2. nem adhatunk utasítást a címkék felsorolására,
3. nem módosíthatjuk a tárolómező elosztását, nem rögzíthetjük újra a programot,
4. lépésként nem hajthatjuk végre a programot, nem rögzíthetjük a Pause billentyűt a végrehajtás ideje alatt.

Megtilthatjuk a gépnek, hogy a védett kártyaoldalt a nem megfelelő tárolótömbbe olvassa be. Ha a védett kártya egyik oldalát beolvassuk a programtárolóba, beiktatódik egy belső programvédő jelző, amely biztosítja a felsorolt korlátozásokat. Az egyetlen lehetőség ezeknek a korlátozásoknak a megszüntetésére (a jelző kiiktatására) a 2nd CP billentyűzése, vagy a készülék kikapcsolása. A programtárolóban levő védett programot átírhatjuk, de a védőjelzőt ezzel nem vonjuk vissza. Az új program ugyanúgy védett lesz, mint az előző.

Az adattárolók tartalmát nem védhetjük. Hasonlóan nem védhetünk olyan tömböt sem, amely a tárolómező elosztása során középre került; ha egy tömb adattárolót tartalmaz, a tömb elején található. Ezért a készülék a teljes tömböt adattárolónak tekint. Nem védett programot olyan kártyára, amelyen előzőleg védett programot tároltunk, úgy rögzíthetünk, hogy a szokásos rögzítési eljárást alkalmazzuk. Az új rögzített program nem lesz védett.

A mágneskártyák olvasása

Ha a kártyát kézi üzemmódban betoljuk az erre szolgáló részbe, a továbbító mechanizmus azt automatikusan áttolja a számológépen. Az, hogy a kártyát elolvassuk-e vagy sem, a kijelzőtároló tartalmától és a tárolómező elosztástól függ.

1. 0 kijelzése esetén bármelyik tömb elolvasható, feltéve, hogy a pillanatnyi elosztás szerint ez a tömb a programtárolóhoz tartozik. Ha az elosztás hibás, a készülék a kártyát nem olvassa el, így a kijelzett tömbszám (amely a beadott kártyára van írva) villog
2. Ha a kijelzőben n van, a készülék csak az n tömböt tudja elolvasni. Ha a kártyán egy másik tömbszámot tárolunk, a kiolvasott szám a kijelzőben villog, és a készülék a kártyát nem olvassa el. Helytelen elosztásnál is villog a beadott tömb száma és szintén nem történik olvasás.
3. Ha a kijelzőben $-n$ áll, minden beolvasott tömb az n tömbbe kerül. Ha egy kártyát beolvastunk, a kártyán tárolt tömbszám a kijelzőbe kerül. Védett kártya (negatív tömbszámmal) csak abba a tömbbe vihető be, amelyet a kártyán megadtunk.

Ha egy kártya beadása után a kijelzőben nulla villog, a készülék beolvasási hibát jelez és a beolvasási folyamat meg kell ismételn.

Ha egy kártya beadásakor más szám vagy $\pm n$ ($n = 1, 2, 3, 4$) van a kijelzőben, a készülék a kártyát nem olvassa be, és a két jobboldali számjegy villog a kijelzőben.

Programozott kiolvasási utasítás

A program végrehajtása közben az INV 2nd Write a számológépet arra utasítja, hogy a fenti szabályokkal összhangban olvasson be egy mágneskártyát. A kártyát betoljuk a készülék kártyaolvasójába; de beolvasás csak akkor történik, ha a programban sorra kerül az INV 2nd Write utasítás. Így szükség esetén adatokat táplálhatunk be.

Ügyeljünk arra, hogy a kártyaolvasóba betolt kártya elolvasása automatikusan megtörténik, ha a programot az INV SBR vagy R/S utasítással megszakítjuk!

A mágneskártyák beolvasása

Kijelzés a kártya beadásakor	Nem védett programú kártyák	Védett programú kártyák
0	A kártyán tárolt információknak megfelelő számú tömb beolvasása, feltéve, hogy a pillanatnyi elosztás megfelel a kártya elosztásának. Helytelen elosztásnál áthalad a kártya, de beolvasás nincs és a kijelző villog.	Mint a nem védett programnál
1, 2, 3, 4	Az adott tömbszámú kártyaoldal beolvasása, a kártyaoldalszám kijelzése. Másik oldal beolvasásakor vagy rossz elosztásnál a kártya áthalad, de beolvasás nem történik. A kijelzőben az áthaladó kártyaoldal száma villog.	Ha az áthaladó kártyaoldal száma a kijelzettel azonos, a beolvasás megtörténik, és a kijelző negatív sorszámot mutat. Másik kártyaoldal beadásakor vagy helytelen elosztásnál áthalad a kártya, de beolvasás nem történik. A kijelzőben az áthaladó kártyaoldal száma villog
-1, -2, -3, -4	A beolvasott kártyaoldalt kényszerűen bevisszük a megfelelő sorszámú tömbbe, tekintet nélkül a védelemre vagy az elosztásra. Védett programot nem lehet tetszőleges tömbbe beolvasni.	A megadott tömbszámú kártyaoldal beolvasása; a kártyaoldal szám kijelzése. A másik oldal beadásakor, ill. nem megfelelő elosztásnál a kártya áthalad, de beolvasás nincs; a kijelzőben az áthaladó kártyaoldal száma villog
Más szám	A kártya áthalad, de beolvasás nincs, a kijelzett két jobboldali szám villog.	Mint a nem védett programnál

Ha egy kártya beolvasási kísérleténél a kijelzett érték villog, a kártya áthalad ugyan, de beolvasás nem történik és mindkét jobboldali egészszám villog. A mágneskártya olvasásakor a készülék kijelzője nem működhet rögzített tizedespontos beállításban. Ügyeljünk arra, hogy hosszabb számítások előtt a telepek fel legyenek töltve, vagy kapcsoljuk a készüléket a hálózatra! A pontos működés érdekében az különösen fontos mágneskártyák beolvasásakor, ill. beírásakor. A kijelzőben villogó 0 arra utal, hogy a kártya beolvasása helytelenül történt, és ismétlésre van szükség.

A mágneskártyák kezelése és gondozása

Vigyázat! A mágneskártyák megsérülhetnek, vagy az információ megváltozhat, ha azokat káros hatás: por, elektromágneses tér (pl. elektromotor, teljesítmény-transzformátor, állandó mágnes stb.) éri.

A mágneskártyák megfelelő kezeléssel mellett tetszőleges ideig megtartják a beírt információt. Az információ nincs kitéve öregedési folyamatnak, és azt csak külső mágneses tér módosítja. Bár a mágnesjelek értéke időben nem csökken, a kártya fizikai tulajdonságai vagy a készülék kártyatovábbító szerkezete károsodhat.

A mágneskártyák kezelése

A kártyák kezelése nagy gondosságot kíván. Sérült, meghajolt vagy karcolt kártya nem használható rendeltetészerűen. Védjük a mágneskártyát a szennyeződéstől (cigarettahamutól, portól, zsírtól stb.)! A kártya attól is szennyeződhet, ha tisztátalan felületre helyezzük. A bőrfelületről a kártyára kerülő zsír megkönnyíti a por és más idegen testek megtapadását. Ügyeljünk arra, hogy egyetlen szennyezett kártyával nemcsak az olvasószerkezetet, hanem a később használandó kártyákat is beszennyezhetjük! Olajos szennyeződés a kiolvasó szerkezet meghibásodását eredményezheti. Ilyen hiba esetén a készüléket szervizben kell megjavíttatni. Az alábbi utasítások biztosítják a mágneskártyák maximális élettartamát.

1. A kártyát lehetőleg csak a sarkánál fogjuk meg!
2. A kártyát mágnesektől és éles tárgyaktól tartsuk távol; az utóbbi a kártya felületi oxidrétegét megsértheti!
3. A használaton kívüli kártyákat műanyag tasakban vagy más megfelelő tartóban tároljuk!
4. A szennyezett kártyákat azonnal tisztítsuk meg!
5. Ne kísérletezzünk erősen sérült kártyák olvasásával!

A mágneskártyák tisztítása

A szennyezett kártyát különleges tisztító vagy oldószer használata nélkül is megtisztíthatjuk. Ne használjuk a tisztításhoz olajalapú folyadékokat és öngyújtóbenzint! A port és a szennyeződést egyszerűen száraz ruhával távolítsuk el! Egyéb szennyeződést meleg vízben oldott folyékony tisztítószerekkel távolítsunk el! A kártyát öblítsük le, és töröljük szárazra!

A mágneskártya feliratozása

A készülékhez tartozó mágneskártyák természetesen üresek. Felületükön rovatok találhatóak; azonosító számok, jelölések számára. A kártyák feliratait célszerű puha ceruzával, lemosható tintával vagy rosttollal készíteni, hogy azok szükség esetén eltávolíthatók legyenek. Ajánlatos azt a fajta rosttollat használni, amelyet diafilmekhez is alkalmazunk.

A mágnesfej tisztítókártyájának alkalmazása

A készülékhez tartozó különleges jelzéssel ellátott mágnesfej-tisztítókártyán oxidréteg helyett csiszolófelület van. E tisztítókártyával távolítjuk el az olvasó-, illetve az írófejről az oxidréteget vagy szennyeződést. A tisztítókártya nem arra szolgál, hogy különféle zavarok esetén használjuk! Gyakori használata megváltoztathatja a fejek tulajdonságait. Az A függelék Zavarelhárítás c. fejezete utasításokat tartalmaz arra, hogyan használjuk célszerűen a fej tisztítókártyát. A tisztítókártyát a többi mágneskártyához hasonlóan a készülék alsó nyílásába kell helyezni, majd a továbbító rendszer áttolja a készüléken. Ha a kártya áthaladása után a kijelző villog, nyomjuk meg a CLR billentyűt! A mágnesfej tisztítókártyát zavar esetén csak egyszer használjuk, takarékoskodjunk annak alkalmazásával! Ügyeljünk arra, hogy a kártyát tiszta állapotban használjuk!

A továbbító görgő tisztítására szolgáló tisztító kártya; mintegy 500 kizolvasás után vagy a kártya egyenetlen továbbítása esetén használjuk. Ilyenkor 1 2nd Write billentyűzést alkalmazunk, és betoljuk a kártyát. Fogjuk meg erősen a kártyát annak keskeny oldalánál, és a továbbító szerkezet bekapcsolt állapotában tologassuk előre és hátra! 3–4 másodperc elegendő a görgő megtisztításához. Húzzuk ki a kártyát, és nyomjuk meg az R/S billentyűt! Ha a továbbító szerkezet a kártya kihúzása után tovább mozog, akkor rövid időre kapcsoljuk ki a készüléket!

A vizsgálókártya alkalmazása

A vizsgálókártyát a készülék működőképességének és az olvasó-író műveletek ellenőrzésére használjuk. A vizsgálókártya nem ellenőrzi és nem érinti a software-egység működését.

A számológép vizsgálata

A vizsgálókártya alkalmazásakor beolvassuk a kártya 1-es oldalát. Nyomjuk meg a CLR billentyűt és toljuk be a kártyát a készülékbe! A kártya beolvasása után a kijelzőben 1-nek kell megjelennie. Nyomjuk meg az E billentyűt! A kijelzőben megjelenő –8888888888 érték arra utal, hogy nincs hibajelenség. Ha a vizsgálathoz a KA–100 sornymatatót használjuk, a kijelzett értéket ki is nyomtathatjuk. A villogó kijelzés azt jelenti, hogy a vizsgálat hibát állapított meg. Ismételjük meg a vizsgálatot, hogy a hiba valóban fennáll-e! Ha igen, végezzük el az olvasó-író vizsgálatot a hiba behatárolása céljából!

Olvasó-író vizsgálat

Ezt a vizsgálatot célszerű a KA–100 sornymatatóval végrehajtani. Olvassuk be a vizsgálókártya 2. oldalát! Ehhez nyomjuk meg a CLR gombot, és toljuk be a kártyát a készülék nyílásába! A kártya beolvasása után a kijelzőnek 2-et kell mutatnia. Ha nyomtatót is alkalmazunk, billentyűzzük GTO 240 2nd List utasítást és ellenőrizzük, hogy a készülék 240...479 tárolóhelyei a 77 kódot tartalmazzák-e! Sornymatató nélkül a vizsgálatot kézzel billentyűzve végezzük. Nyomjuk meg a GTO 240 LRN billentyűt, és ellenőrizzük az SST utasítással, hogy minden tárolóhelyen a 77 kód található-e! Ha a 240...479 tárolóhelyeken a 77-től eltérő kód van, az a kártyaolvasó hibája.

A beírás ellenőrzésére a kártya 2. oldalát olvassuk be a fentiek szerint! Ezután előveszünk egy üres mágnes-kártyát, CLR2 2nd Write utasítást billentyűzünk, és az üres kártyát betoljuk a készülék nyílásába. Nyomjuk meg a 2nd CP billentyűt, és olvassuk be a készülékbe az éppen beírt kártyát! Vizsgáljuk meg, hogy a 240...479 tárolóhelyeken a 77 kód van-e! Hiba esetén ismételjük meg az egész vizsgálatot, hogy a hiba-jelenség valóban fennáll-e!

A FÜGGELÉK — KARBANTARTÁS ÉS SZERVIZ

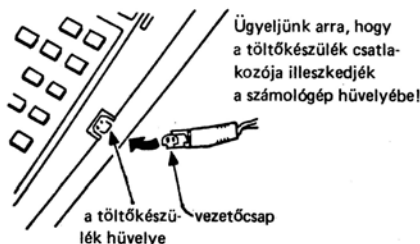
Telepes és hálózati üzem

Szokásos üzemmód:

A hálózattól független üzemeltetés céljából csatlakoztassuk az E-07A töltőkészüléket a 220 V-os 50 Hz-es hálózathoz, 8–10 órán át! A töltőkészülék és a telep hálózati üzemen melegedhet, ez általános jelenség, nincs káros hatása.

Vigyázat: a készülék károsodhat, ha a töltőkészüléket csatlakoztatjuk ugyan, de a telep nincs bekötve!

A töltőkészülék csatlakoztatását a készülékhez az alábbi ábra mutatja.



Teljesen feltöltött telep esetén a készülék kb. 2...3 órán át működőképes. Gyenge telep esetén haladéktalanul kezdjük meg a töltést; mert a készülék hibásan működhet! A kimerült telep jele a gyenge, ellenőrizhetetlenül világító kijelző vagy a kártyatovábbító szerkezet elindulása. Ha beolvasási művelet közben merül ki a telep, a kártyán levő információk törölhetnek vagy megváltozhatnak.

A nikkel-kadmium cellák élettartamát nehéz előre megjósolni, de szokásos üzem esetén az élettartam 2–3 évre, ill. 500–1000 feltöltésre tehető.

Periodikus feltöltés.

Bár a számológép a bekapcsolt töltőkészülékkel tetszőleges ideig működik, a feltöltött telep idővel elveszíti tárolókapacitását, ha nem sütjük ki rendszeresen. A telep élettartamának biztosítására ajánlatos a számológépet havonta legalább kétszer hálózattól függetlenül üzemeltetni! A telep kisülése után természetesen újra fel kell tölteni.

A telep teljes kimerülése

Ha a készüléket hosszabb ideig bekapcsolva hagyjuk, a telep teljesen kisül. Ekkor legalább 24 órán át csatlakoztassuk a töltőkészüléket a kikapcsolt számológéphez! Ha ezután sem áll helyre a telepes üzem, a telepet ki kell cserélni. Az ismételt teljes kisütés erősen csökkenti a telep élettartamát.

Tárolás

Ha a készüléket több héten át nem használjuk, akkor a telepet fel kell tölteni. A telep védett a folyadék kifolyásával szemben.

Hibaelhárítás

1. hibajelenség: a kijelzőben hibás eredmények vagy értelmetlen jelek láthatók, a kijelző sötét, vagy a kártya-olvasó folyamatosan működik.

Elhárítás: fel kell tölteni a kimerült telepet (lásd a Telepes és hálózati üzem c. fejezetet a függelék elején).

2. hibajelenség: a kijelző sötét marad.

Elhárítás: rövid ideig tartsuk lenyomva az R/S billentyűt! Ha utána ismét kijelzést látunk, a készülék egy hosszú programot dolgoz fel, hurokban van, vagy egy mágneskártya betolását várja.

Tartsuk rövid ideig benyomva az RST billentyűt! Ha ismét kijelzés következik, a készülék a software-programot dolgoz fel, vagy hurokban van. Az is lehet, hogy a telep majdnem kimerült.

3. hibajelenség: billentyűzés közben a kijelző villog.

Elhárítás: nem megengedett műveletet vagy utasítást írtunk be, vagy túlléptük a készülék kapacitáshatárait. A feltételek felsorolása a B. függelékben található.

4. hibajelenség: a kijelző villog, ha software-programot hívunk be.

Elhárítás: ilyen számú software-program nincs (lásd a software-programok kézikönyvét); esetleg a software-egységet rosszul tettük be (lásd e kezelési utasítás III. fejezetét).

5. hibajelenség: a kijelző villog vagy helytelen eredményt mutat, ha software-programot hajtunk végre.

Elhárítás: lehet, hogy hibás programbehívás történt, esetleg hibás a programozás. Lásd a programozásra vonatkozó részt a kézikönyvben!

A programelosztás során kevés az adattároló, a program végrehajtására.

Nyomjuk meg az INV 2nd Fix billentyűt, vagy rövid időre kapcsoljuk ki, majd ismét be a készüléket, és újra hajtjuk végre a programot!

Nyomjuk be a CLR 2nd Pgm 1 SBR = billentyűket, és futtassuk le a software-vizsgáló programot! Ha a kijelző villog, vizsgáljuk meg, hogy a software-egységet helyesen tettük-e be (lásd a III. fejezetet), majd ismételjük meg a billentyűzést!

6. hibajelenség: a kijelző villog, vagy helytelen eredményt mutat, ha saját programot futtatunk.

Elhárítás: a program során nem megengedett művelet, túlcsoordulás vagy alácsoordulás fordult elő. A hibakeresésben segítséget nyújt a B, C, D függelék. Lehetséges, hogy egy software-programot hívtunk be, ekkor nyomjuk meg az RST billentyűt és kezdjük az egészet előlről!

Nyomjuk meg a CLR 2nd Pgm 1 SBR 1 billentyűt, a software-vizsgáló program végrehajtására! Ha a kijelző villog, a hibaelhárítást az 5. pont szerint végezzük! Ha a programot mágneskártyáról olvastuk be, végezzük el az ellenőrzést a vizsgálókártyával (lásd: utolsó fejezet)!

Ha a vizsgálati eredmények nem mutatnak hibát, vizsgáljuk meg a mágneskártyát saját programunkkal, vagy ellenőrizzük, hogy nem szennyezett-e! Lásd a hibaelhárítással kapcsolatban a 7. pontot!

7. hibajelenség: a kijelző villog, a mágneskártya beírása vagy beolvasása közben.

Elhárítás: rosszul végeztük a műveletet (lásd a VII. fejezetet), vagy helytelenül választottuk meg a tárolók elosztását!

Ha a többi kártya jól kiolvasható, nézzük meg, hogy az első kártya nem szennyezett vagy sérült-e! Szükség esetén tisztítsuk meg vagy cseréljük ki a hibásat! Ha a kártya szennyezett volt, végezzük el az utolsó fejezetben leírt mágnesfejtisztítást és hajtógörgő-tisztítást! Ha más kártyát sem tudunk helyesen beolvasni, használjuk egyszer a mágnesfejtisztító kártyát!

8. hibajelenség: a számológép nem kapcsolható tanulási üzemmódba, nem léptethető lépésenként, nem sorol fel és nem rögzít kártyára.

Oka: a program védett. (Lásd a programok adatvédelméről szóló részt!)

A készülék javításra való beküldése esetén mellékelni kell a töltőkészüléket, a software-egységet, és mindazokat a mágneskártyákat, amelyeknél meghibásodások mutatkoztak. A készüléket gondosan becsomagolva küldje be a szervizbe!

A számológép szervíz címe:

Budapest VII., Szövetség u. 20. Telefon: 422-972.

Ügyfélfogadási idő: hétfőtől–péntekig 9–14 óra.

A szervíz a PTK–1096 típusú számológéphez a szavatossági időn túl 7 évig biztosít cserealkatrészt.

A készülék javításával, működésével vagy különleges alkalmazásával kapcsolatos kérdésekkel forduljon a Híradástechnika Szövetkezet számológép szervizéhez!

B. FÜGGELÉK – HIBAFELTÉTELEK

A villogó kijelzés arra utal, hogy a készülék kijelzőkapacitását nem vettük figyelembe vagy a készüléknek nem megengedett utasítást adtunk. A villogást a CE vagy a CLR utasítással állíthatjuk le. A CLR utasítás a kijelzést és a befejezetlen műveleteket is leállítja, illetve törli. A CE utasítás csak a villogást állítja le, de további számítások zavartalanul végezhetők a befejezetlen műveletekkel. A kijelző villogásának okai a következők:

1. Valamely beírás vagy eredmény a kijelzőben vagy az adattárolókban túllépi a készülék kapacitását, vagyis kivülesik a 0 és $\pm 1 \cdot 10^{-99} \dots \pm 9,9999999 \cdot 10^{99}$ tartományon. A tartomány túllépése esetén a kijelző villog, ez mutatja a túlszordulást, vagy alácsordulást.
2. Árkuszfüggvény nem megengedett függvényértékkel (pl. $\arcsin x$; $x > 1$ esetén).
3. Negatív szám logaritmus. Az előjelhiba miatt a szám abszolút értékének logaritmus. Villog a kijelzőben.
4. Negatív szám hatványának vagy gyökének számítása. A szám abszolút értékének hatványa vagy gyöke villog a kijelzőben.
5. Közvetlenül egymás után két műveleti billentyűt nyomtunk meg. Ez érvényes a +, -, X, ÷, y^x, és $x\sqrt{y}$ utasításra, amikor is az utoljára beírt szám villog.
6. Az = vagy a) billentyűt közvetlenül a +, -, X, ÷, y^x vagy $x\sqrt{y}$ billentyű után nyomtuk meg, vagy egy szám után közbenső művelet nélkül azonnal (billentyűzése következik. Ilyenkor az utoljára beírt szám villog a kijelzőben.
7. 9-nél több zárójelet nyitottunk, vagy 8-nál több művelet maradt befejezetlen. A 10. zárójelet vagy a 9. műveletet a készülék nem veszi figyelembe, tehát a feldolgozás folytatható. Az utoljára beírt szám villog a kijelzőben.
8. Egy szám osztása 0-val. A kijelzőben 9.999999 99 villog.
9. Különleges vezérlőművelet behívása a 00–39 tartományon kívül.
10. Rajzolás (Op 07) a 00...19 tartományon kívül. A kijelzett érték villog.
11. Elosztáson kivüleső, nemlétező tárolóhelyekhez vagy nem kijelölt címkehez próbálunk elágazást létesíteni.
12. Software-programot akarunk átvinni a programtárolóba, melyhez nincs elegendő tárolókapacitás.
13. Nemlétező software-programszám címzése. A kijelzett pillanatnyi érték villog.
14. Villog a kijelző lineáris regresszióval kapcsolatos számítások esetén a következő esetekben: ha a regressziós egyenes az y tengellyel párhuzamos, és a meredekséget, az y tengellyel való metszéspontot,

a korrelációt, az x' vagy az y' értékét akarjuk kiszámítani. Ha az egyenes az x tengellyel párhuzamos, a kijelző akkor villog, ha az x' értékét vagy a korrelációt számítjuk.

- Kettőnél kevesebb adatpont beadása esetén, ha a meredekséget, a tengelymetszést, a korrelációt, az x' vagy az y' értékét akarjuk számítani, az utolsó kijelzett érték villog.
- Négynél több műveletet hagyunk befejezetlenül a gépben, s közben lineáris regressziót, trend-elemzést, statisztikai műveletet, poláris-derékszögű vagy fok-perc-másodperc átszámítást végzünk.
- A 0^{-x} és a $^{-x}\sqrt{0}$ műveletnél túlsordulást („9.9999999 99”) jelez a készülék.
- A poláris-derékszögű átszámításoknál az r távolság értéke a $10^{\pm 50}$ tartományon kívül fekszik. A kijelzőben a szögérték villog.
- Ha a független változó értéke a megadott értelmezési tartományokon kívül esik, akkor a kijelzés villog.

Függvény	Értelmezési tartomány
$\arcsin x, \arccos x$	$-1 \leq x \leq 1$
$\ln x, \log x$	$1 \cdot 10^{-99} \leq x < 1 \cdot 10^{100}$
e^x	$-227,9559242 \leq x \leq 230,2585092$
10^x	$-99 \leq x < 100$

20. A $\pm 90^\circ$, a $\pm \pi/2$ rad ill. a ± 100 újfok páratlan számú többszörösei nem értelmezhető változói a tangensfüggvényeknek; akkor is helytelen eredményre vezetnek, ha hibajelzés nem történt. (Lásd a C. függelékét!)

21. A mágneskártyák helytelen beolvasása vagy beírása esetén is villog a kijelző.

Hibák a program végrehajtása közben

Ha a fenti hibák valamelyike a programban előfordul, a további döntés a kezelő feladata. A hibafeltételtől nem következik okvetlenül programmegszakítás. A program egyébként folytatódik, és azt a hibás értéket használja a további számításokhoz, amely a számítási üzemmódban villogott. Ha a program áll, a hibát villogó eredmény jelzi. Hogy ez az eredmény helyes-e vagy sem, a feladattól és hibafeltételtől függ. Hibafeltétel fellépésekor a programozó megszakítja a programot. Ehhez beiktatja a 8. jelzőt, vagy alkalmazza az Op 18 és az Op 19 hibavizsgáló utasítást.

C. FÜGGELÉK – A KIJELZETT EREDMÉNYEK ÉS PONTOSSÁGUK

A számológépnek más elektronikus és mechanikus készülékekhez hasonlóan, meghatározott szabályokkal és megadott értéktartományban kell dolgoznia.

A készülék pontosságát a számításokhoz használt számjegyek száma határozza meg. A számológép látszólag a kijelző szerint 10 számjeggyel dolgozik, de a számításokat valójában 13 számjeggyel végzi; a 3. ún. védőhely a pontosságot javítja. A beépített 5/4-es kerekítés biztosítja a 10 jegyű kijelzést. Tekintsük pl. a következő példát, ha ezek a védőhelyek hiányoznak:

$$\frac{1}{3} \cdot 3 = 0,999\ 999\ 999$$

A példa helytelen eredményt ad. Egy 13-jegyű 9-es sornál a 10 helyre való kerekítés 1-et ad eredményül.

A magasabbrendű matematikai függvényeknél ismétléses számításokat használunk. Általában a kerekítési hiba felhalmozódik. Egy szám 13 jegyű ábrázolása a kijelzett 10-jegyűtől 3 nagyságrenddel különbözik. Így biztosítjuk, hogy az eredményeket a készülék a 10. jegyre pontosan kerekítse.

Általában ezeket a védő számjegyeket teljesen figyelmen kívül hagyhatjuk, de bizonyos számításoknál váratlan eredmények mutatkozhatnak. A véges műveletek határa, a megszakítási és a kerekítési hiba

nem biztosítja minden esetben a védőhelyeknek megfelelő teljes pontosságot. Ezért a készülék két matematikailag azonos függvény kivonása esetén 0-tól eltérő eredményt adhat.

$$\text{Pl. } \sin 45^\circ - \cos 45^\circ = 0.$$

Billentyű	Kijelzés
45 2nd sin —	0.7071067812
45 2nd cos	0.7071067812
=	7.—13

A $\sin 45^\circ$ és a $\cos 45^\circ$ azonosan kijelzett eredményei azt mutatják, hogy a függvények legalább 10 számjegyre pontosak. A végeredmény eltérése csak a 13. hely pontatlanságának következménye. Ezért a 10^{-11} alatti különbség gyakorlatilag 0-nak tekinthető.

Ennek a ténynek akkor van jelentősége, ha magunk írjuk a programot. Ha megvizsgáljuk a számított eredményt, hogy az egy másik értékkel egyenlő-e (pl. $x = t$ utasítás), vigyázni kell, hogy a védett helyeken levő eltérések miatt ne kapjunk helytelen eredményt! Az $EE \text{ INV } EE$ utasítás leválasztja az eredményről a védett jegyeket és csak kerekített és kijelzett értékkel számol tovább a készülék.

Ha a B. függelékben felsorolt korlátokat nem lépjük túl, alapkijelzés esetén az összes számított eredmény pontos. Kivételek a következők:

Szögfüggvények

Alapkijelzésnél az eredmény hibája a 10. számjegynél ± 1 lehet, a $\pm 36\,000$ fok, a ± 200 radián vagy $\pm 40\,000$ újfok tartományban. Ha x értéke a $\pm 3,6 \cdot 10^{14}$ fok, $\pm 6,2799993 \cdot 10^{12}$ radián vagy $\pm 4 \cdot 10^{14}$ újfok, a függvények periodicitása már nem ismerhető fel. Általában a pontosság a megadott határértékeken kívül dekádoként egy számjeggyel csökken. Kivételt képez a tangensfüggvény, a $\pm 90^\circ \pm \pi/2$ rad, ill. ± 1009 páratlan többszöröseinél, ami túlsorduláshoz vezet, mert a függvény ezekben a pontokban nincs értelmezve.

Hatványok és gyökök

A pontosság a hatványok és a gyökök számításakor csökken, ha az y hatványalap igen közel kerül az 1-hez, és az x hatványkitevő igen nagy értékű.

Pl. a $0,99999944^{-1\,60\,000}$ 8 kijelzett számjegyre pontos, míg a $0,99999944^{-4\,00}$ mind a 10 kijelzett számjegyre pontos.

D. FÜGGELÉK – HIBAKERESÉS A PROGRAMOKBAN

Még a tapasztaltabbak is kerülhetnek olyan helyzetbe, hogy a program nem működik helyesen. A készüléknek van egy olyan billentyűcsoportja, mely megkönnyíti a javítást. Természetesen először meg kell vizsgálni, hogy a programot helyesen írtuk-e be. Ha itt nem találunk ellentmondást, akkor ez a függelék biztosan segíteni fog. A következőkben összeállítottunk egy jegyzéket a leggyakoribb programozási hibákról. Ezek alapján megvan a lehetőség a hibák felismerésére.

A hibakeresésben nagy jelentősége van a KA-100 típusú sornyomtatónak a programjegyzék felsorolásánál, illetve a számítások jegyzőkönyvezésénél.

Alapvető szempontok

Algebrai műveleti rendszer

A legtöbb feladatot olyan sorrendben billentyűzhetjük be a készülékbe, ahogy azt leírjuk; ez azonban nem jelenti azt, hogy a készülék ugyanilyen sorrendben értelmezi is. A számológép algebrai rangsora szerint a $2 + 3 \cdot 6$ kifejezést úgy értelmezi, mint $(3 \cdot 6) + 2 = 20$, és nem $(2 + 3) \cdot 6 = 30$. Az egyváltozós függvények esetében az utasításnak a független változó értékét kell követnie. A $\sin \pi$ értékét például a $2\text{nd } \pi$ 2nd sin billentyűzéssel számítjuk ki.

Egyenlőségutasítás (=)

Az egyenlőségutasítás minden befejezetlen műveletet lezár. Ezért különösen alprogramokban gondoljuk meg annak alkalmazását! Az egyenlőségutasítás hiánya is zavarokhoz vezethet. Tekintsük pl. a $(2 \cdot 3)^2$ kifejezést! A $2 \times 3 \times 2$ utasítással a készülék a $2 \cdot 3^2$ értéket számítja ki. A helyes billentyűzés a következő:

$$2 \times 3 = x^2 \text{ vagy } (2 \times 3) x^2$$

Befejezetlen műveletek

A számológép 9 zárójelet és 8 befejezetlen műveletet képes feldolgozni. Néhány billentyű azonban 4 befejezetlen műveletet igényel. Ilyenek a poláris-derékszögű és a szögérték-átszámítások, valamint a statisztikai műveletek billentyűi.

A fenti határokat túllépő beírást a készülék nem veszi figyelembe, és a kijelzés villog.

Többszörös címkek

Egy címkét egy programban csak egyszer szabad felhasználni. A címkekereső rendszer mindig a 000 tárolóhelyen kezd és nem azon a helyen, ahol a címkét behívtuk. Ezért az átmenet mindig az első címkehez történik, második alkalmazásánál ez a címke már nem található meg.

Alprogram

A 6 alprogramsík valószínűleg több, mint amennyire bármikor is szükség lehet. Ha egy, a hatodik síkon kívül levő alprogramot hívunk be, a készülék nem tárol új visszaléptetési címet az alprogram-visszaléptető tárolóban. Ha az INV SBR utasítást a 7. alprogramsíkon találjuk, a feldolgozás az utolsó, ill. hatodik címnél folytatódik, amely a visszaléptető tárolóban van, és természetesen nem jut el a kívánt helyre. Az azonban nem okoz hibajelzést. Szigorúan ügyelnünk kell tehát arra, hogy ne következzen be hibás elágazás! Gondoljuk meg, hogy egy programcím-billentyű alprogrambehívás, ha nincs előtte elágazási utasítás (pl. GTO vagy $2nd\ x = t$)! A poláris-derékszögű átszámítások, a szögérték-átszámítások, valamint a statisztikai műveletek is egy alprogramsíkot foglalnak el.

A reset utasítás (RST)

Az RST utasítás alkalmazásakor figyelembe kell venni annak valamennyi rendeltetését, hogy a nem kívánt hatásokat is elkerüljük. Az RST utasítás rendeltetése: a programszámláló visszaállítása a 000 tárolóhelyre, az összes programjelző kiiktatása, az alprogram-visszaléptető tárolók törlése, a software-program megszakítása, és a programszámláló visszaállítása a programtárolóra.

Statisztikai számítások

Előre programozott statisztikai számítások esetén a készülék az adatokat az R1..R6 tárolókban összegzi. Ezért a statisztikai számításokkal készített programnál ezeket az adattárolókat nem szabad lefoglalni, azokat az adatbeírás előtt törölni kell. Ügyelni kell arra is, hogy a t-tárolóban levő érték megmaradjon, mert arra a programban később szükség lesz! Mint már említettük, ezek a függvények négy befejezetlen műveletet és egy alprogramsíkot igényelnek.

Poláris-derékszögű átszámítások

Itt elsősorban a szövegységek helyes megválasztására kell gondolni. A számítások egy alprogramsíkot igényelnek, és max. 4 befejezetlen művelet van.

A szövegység meghatározása

Bekapcsoláskor a készülék (régi) fokra van beállítva. Ha a szögértéket radiánban vagy újfokban akarjuk kifejezni, akkor a megfelelő billentyűket kell használni. A készülék a választott szövegységben számol, (amíg a má sikra át nem állítjuk) de szövegység-átszámítást nem végez.

Függvények, amelyek csak a kijelzett értékkel számolnak

Az EE és a 2nd D.MS műveletnél a további számítás a kijelzett értékkel folytatódik. Ez azt jelenti, hogy ezen utasítások alkalmazásakor a rögzített tizedesponntal biztosított védőhelyek elvesznek.

Összehasonlítás a t-tárolóval

Annak eldöntésére, hogy egy új programtárolóhely felé létrejött-e az elágazás vagy sem, a kijelzőtároló és a t-tároló tartalmát a $2\text{nd } x = t$ vagy a $2\text{nd } x \geq t$ utasítással hasonlítjuk össze. Kísérletezzünk példaként a következő billentyűzéssel!

2nd Deg 45 2nd sin $x \approx t$ 45 2nd cos 2nd $x = t$ 114

Ha most lenyomjuk az L.R.N. billentyűt, az elágazás nem jön létre, bár a $\sin 45^\circ$ és a $\cos 45^\circ$ matematikailag egyenlő. Ennek oka az, hogy a készülék a tartalékhelyeket kerekíti, amikor ezeket az értékeket számítja. (Lásd a C. függelék.) Vonjuk ki a $\cos 45^\circ$ -ot, hogy erről magunk is meggyőződjünk! Nullától eltérő eredményt kapunk, mivel a védőhelyeken levő értékek eltérők. Általában ezek az eltérések lényegtelenek, feltételes elágazásoknál azonban figyelembe kell venni. Az EE INV EE utasítás leválasztja a védőhelyeket az eredményről, és a további számoláshoz csak a kerekített érték marad meg.

Vigyáznunk kell az EE és a 2nd D.MS műveleteknél, mert ezeknél a védőhelyek törölődnek vagy megváltoznak!

A program változtatása

Programok változtatását gondosan végezzük! A legegyszerűbb változtatásokat, amelyek egyik helyen jelentéktelennek látszanak, a másikon hibát okozhatnak. A változtatás minden következményét vegyük figyelembe! Különös figyelmet kívánnak az összevont címek, a hasonló címkek, az összevont utasítások (pl. INV SBR), továbbá az olyan címek, amelyeket billentyűkóddal jelölünk. Védett programot nem javíthatunk, illetve nem módosíthatunk. Gondoljunk arra, hogy a programlépések felülről lefelé változtatlanul eltolódnak, ha utasításokat iktatunk be, vagy törölünk! Az abszolút címzésű elágazási utasításokat tehát ennek megfelelően kell módosítani.

A tárolómező elosztása

Ügyeljünk az adattárolók és a programtárolók megfelelő elosztására! A tárolómező elosztását a 2nd Op 16 billentyűvel ellenőrizzük!

Programvizsgálat

A fent megadott pontok figyelembevételével adunk néhány javaslatot a programvizsgálathoz. Ezek a hibás programok kiértékelését teszik lehetővé.

A program nem zárul le

Ha a program végén a futás nem fejeződik be, akkor a program egy végtelen hurokban fut.

A legjobb eljárás a program lépésenkénti gondos elemzése és az egyes utasítások felülvizsgálata, különös figyelemmel az elágazási utasításokra. Bár az ok valószínűleg egy feltételes elágazás, mégis vizsgáljuk meg először a feltétel nélküli átmeneteket, mert ezeknél a hiba könnyebben megállapítható. Ügyeljünk mindeinek előtt az olyan utasításokra, mint a 2nd Lbl D...GTO D! A hurok megszakításához itt egy feltételes elágazás szükséges. Következő lépésben vizsgáljuk meg a feltételes elágazásokat, különösen ha azokat a hurok elejére vagy végére programoztuk!

A 2nd Dsz utasítás nem küldheti végtelen hurokba a programot; hacsak a program nem hajt végre olyan műveletet, ami a csökkentendő adattároló tartalmát megváltoztatja, vagy 10^{10} fölé emeli. Állapítsuk meg, hogy ez az adattároló nullára állítható-e! Ha a csökkentendő értékű adattárolóban igen nagy számot tárolunk, a program a lezárásig rendkívül hosszú ideig fut, és ez végtelen huroknak tűnik. Ha a programnak meg kell határoznia a szükséges hurok számát, ezt a számítást célszerű felülvizsgálni.

Hasonlóan felül kell vizsgálni a t-tároló összehasonlításával járó feltételes elágazásokat. Ha az ilyen típusú elágazásokat hurokvezérlésre használjuk, általában azt várhatjuk, hogy számításaink előre megadott hatá-

rokhoz tartanak. Ellenkező esetben a huroknak nem lesz vége. Vizsgáljuk meg az egyenleteket, valamint azokat az utasításokat, amelyekkel az egyenleteket programoztuk! Bizonyosodjunk meg arról is, hogy helyes vizsgálóértéket tároltunk-e a t-tárolóban! Ha a programnak ezt a számot kell meghatároznia, vizsgáljuk meg ezeket a számításokat is! További gondot jelenthet az, hogy ezek az utasítások a t-tároló teljes tartalmát összehasonlítják a kijelzőtároló teljes tartalmával; ha csak nincs közben olyan döntés, amely egy új tárolóhelyhez vezet. (Lásd az összehasonlítás a t-tárolóval c. részt és az V. fejezetben a feltételes elágazásokat!)

Ha egy software-programot alprogramként alkalmazunk, megnyomhatjuk az RST billentyűt. Ha a program most megszakad, a software-program végtelen hurokban futott. Tájékozódásul lapozzuk fel a megfelelő software-program utasításait! Az RST utasítás „szükségintézkedés”, és csak a végső eszköz, mert a leállás helyét nem lehet előre megmondani, és a kijelzett érték nem azonosítható.

Ha a kiértékelés után sem találunk hibát, olvassuk el A számológép alkalmazása a hibakeresésnél c. részt!

Ellentmondásmentes adatok ellentmondásos eredményt adnak.

Ezt a hibát általában olyan feltételes elágazások okozzák, amelyek hibás alkalmazáskor egyszer helyes, más-kor viszont helytelen eredményt adnak. Ennek szemléltetésére vizsgáljuk a következő példát!

```

2nd Lbl
A
INV
x>=t
B
2nd St flg
3
2nd Lbl
B
.
.
.
2nd If flg
3
C
.
.
.
2nd Lbl
C
.
.
.

```

Itt a programozó át akarja lépni a program egy részét, ha az eredeti adatok 0-nál kisebbek. Tegyük fel, hogy a következő adatsortot írjuk be a programba: 12, -16, 12. A program az első két beírásnál helyes adatokat szolgáltat. Ha azonban a 12-t másodszor is beírjuk, az eredmény hibás. Ennek oka az, hogy a -16 beadáskor a 3. jelzőt léptettük. Mivel a program nem tartalmaz olyan utasítást, amely a 3. jelzőt visszaléptetné, pozitív beírás esetén a 12-t a második esetben a készülék negatív számként kezeli. Ez esetben a hibát úgy háríthatjuk el, hogy a 2nd Lbl B után az INV 2nd St flg 3 utasítást programozzuk.

Hasonló gondok léphetnek fel minden elágazási műveletnél. Sajnos nincs arra lehetőség, hogy minden egyes esetre példát adjunk.

Állandóan helytelen eredmények

Lehetséges, hogy egy program megírása hibásan történt. Ilyenkor a készülék mindig ugyanazokat a hibás eredményeket adja. Ha az egyenletek kézi billentyűzésű vizsgálatokor nem találunk hibát, akkor tekintsük a következő szakaszt.

A számológép alkalmazása hibakeresésnél

Ha meghatároztuk, hogy milyen értékeket kell kiszámítani és kijelezni és azokat a feldolgozás különböző szakaszaiban hol kell tárolni, akkor a számológépet hatásos eszközként használhatjuk a hibás programok felülvizsgálatára.

Egy fontos tanács: ha egy bizonyos vizsgálati ponton előhívjuk az egyik adattároló tartalmát, a futás folytatása előtt győződjünk meg arról, hogy a kijelzőtároló tartalmát újra visszavittük-e a tárolóba! Egyébként látszólagos hibák léphetnek fel. Ajánlatos az adatokat felülvizsgálat céljából a 2nd Exc utasítással behívni a kijelzőbe, és azokat ugyanazzal a billentyűzéssel helyettesíteni. Így módon az utoljára számított érték kerül ismét a kijelzőbe.

A program futás közbeni elemzésére több utasítás létezik. Az R/S utasítás beiktatása a programba a különböző kulcspontokon igen gyors módszer annak felderítésére, hogy egy hiba hol fordul elő először. Ha a program megszakad, ne csak a kijelző tartalmát, hanem az adat- és a t-tárolót is ellenőrizzük!

Az R/S utasítás beiktatásakor az a legegyszerűbb, ha a legmagasabb sorszámu programtárolóhelynél kezdünk, és visszafelé haladunk. A R/S utasítások beírása fordított sorrendben nem tolja el a tárolóhelyeket, de így is hozzáférhetünk ezekhez az utasításokhoz.

Ha elletmondást találunk, hajtuk végre újra a programot, és szakítsuk meg az R/S utasítással ott, ahol hibát találtunk! Ezután lépésenként folytassuk a program végrehajtását, az SST billentyűvel addig, amíg a hiba pontos helyét meg nem találtuk. A hiba azonosítása és javítása után rendre töröljük az R/S utasításokat! Folytassuk a folyamatot addig, amíg az összes hibát fel nem számoltuk!

Az SST billentyű helyett használható a lenyomott GTO billentyű is. Ezzel az egyes lépések közé rövid szünetet iktatunk be és követhetjük a program lefutását. Ennél a módszernél először el kell indítani a programot. Annak biztosítására, hogy az összes programutasítást kezdettől megfigyelhessük, tegyük az alábbiakat:

1. az R/S billentyűt megnyomjuk és lenyomva tartjuk,
2. közben nyomjuk be a GTO billentyűt és tartuk benyomva,
3. az R/S billentyűt ismét engedjük el!

A vizsgálat akkor a legegyszerűbb, ha a feladat hibafeltételhez vezet. Ebben az esetben léptessük a 8. jelzőt és a programot újra hajtuk végre! Ha hibahelyhez érünk, a program megszakad. Az LRN billentyű segítségével feljegyezzük az a programtárolóhelyet, ahol a hiba fellépett. Valójában a feldolgozás akkor szakad meg, amikor a programszámláló a hiba utáni első helyre mutat. Ezután a hiba jellegét meghatározhatjuk, és a szükséges javítást elvégezhetjük.

Vigyázat! A software-program lépésenként nem hatítható végre. Ha azon a programtárolón, amely egy software-programot hív be, lépésenként végigmegyünk, a software-program lefutása alatt a kijelző sötét marad. Ennek lezárása után folytathatjuk a főprogram végrehajtását, az SST vagy a GTO utasítással. Ha a GTO utasítást alkalmazzuk, akkor a billentyűt nem szabad elengedni a software-program végrehajtása alatt, hogy a főprogram egyetlen része se kerülje el a vizsgálatot!

A KA-100 sornyomató alkalmazása hibakereséshez

A sornyomató értékes segédeszköz egy program hibameghatározásánál. Ha a számológépet rákapcsoltuk a sornyomatóra, a következők a teendők.

1. Nyomjuk meg az RST 2nd List billentyűt; ezzel megkapjuk a programutasítások, tárolóhelyszámok, utasítás kódok és utasítástárolók teljes felsorolását. Így főlölegessé válik a program lépésenkénti felülvizsgálata.
2. Hajtuk végre a programot, és a nyomtatót kapcsoljuk párhuzamos üzemmódba! A számításokat lépésről lépésre követhetjük, és pontosan megállapíthatjuk, hogy hol tér el a program a kívánt megoldástól.
3. Nyomjuk meg az XX INV 2nd List billentyűt, mire megkapjuk az R_{xx}-től kezdve az összes adattároló tartalmát. Ha a programot különböző helyeken megszakítjuk, és ezt a műveletet végrehajtjuk, könnyen megvizsgálhatjuk, hogy az adattárolók a helyes adatokat tartalmazzák-e.
4. Ha megnyomjuk az RST 2nd Op 08 billentyűt, a készülék kinyomtatja az összes címke, továbbá azok abszolút címeit. Ezzel az utasítással nem kell a teljes program-felsorolásból kikeresni a címkek helyzetét.



Gyártja: HIRADASTECHNIKA
1400 Budapest, Pf. 23,
Telek. 22-6151 (ntsz/h)