

PTK-1023 SCIENTIST PR
KEZELÉSI ÚTMUTATÓ



A TK-1023 KEZELÉSI ÚTMUTATÓVAL EGYÜTT ALKALMAZANDÓ!

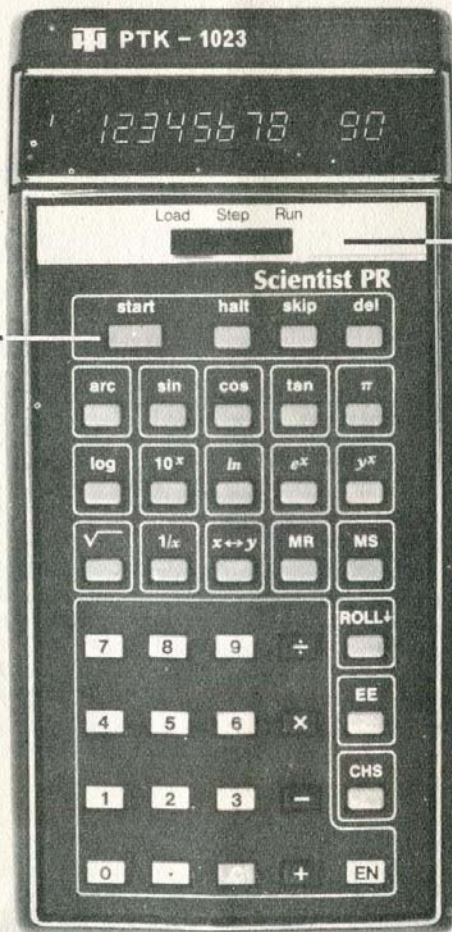
BEVEZETÉS

Alapfunkcióit tekintve a PTK-1023 megegyezik a TK-1023-as számológéppel, mivel ebbe a készülékbe is ugyanaz az aritmetikai egység került beépítésre. A két készülék abban különbözik egymástól, hogy a PTK-1023-ba beépített program egység lehetővé teszi, hogy a készülék 100 programlépésig összetett műveletek megoldására is programozható legyen.

Ez a kezelési útmutató nem tárgyalja az alapfunkciókkal kapcsolatos tudnivalókat, mivel azokat a mellékelt TK-1023-as kezelési útmutatóban adtuk meg, így kizárólag csak a programozhatósággal foglalkozik.

Program vezérlő billentyűk

Program üzemmód kapcsoló



PROGRAMOZHATÓSÁG

A gép programozhatósága lényegében a billentyűk automatikus működtetési lehetősége. A programozhatóság fő előnye a program elkészítésének és felhasználásának egyszerűségében rejlik. A programozási lehetőség kihasználásához sem különleges ügyesség, sem a programnyelvek ismerete nem szükséges. Ha a gép használója a számológéppel tud számolni, akkor a programozó berendezést is tudja használni!

A programozhatóság által az egyébként is igen nagy teljesítményű A PTK-1023 számológép hosszú számítási műveletek és iterációk (ismétlődéses) számítási feladatok megoldására képes és igen nagy időmegtakarítást biztosít használatjának.

A gép programozhatóságának lényege, hogy a gép a számítási feladat megoldása során működtetett billentyűk működtetésének sorrendjére emlékszik. Ezért a gép használójának a programozáshoz semmi egyebet nem kell tennie, csak a program kapcsoló LOAD állásában a feladatot egy ízben beírni. Ha a beíráskor hibát követünk el, a hiba kijavítható és a gép végül a helyes eredményt fogja adni.

A programozás kezelőszervei

LOAD/STEP/RUN üzemmód kapcsoló: a program beírásának, léptetésének, végrehajtásának kapcsolója

LOAD — a kapcsolónak ebben az állásában írhatjuk be a programlépéseket a számológép programtárába

STEP — ha a kapcsoló STEP állásban van, a gép a start billentyű minden egyes lenyomására a tárolt program egy-egy lépését végzi el

RUN — ha a kapcsoló RUN állásban van, a gép a start vagy a skip billentyű lenyomására a teljes programot lefuttatja.

A programozás billentyűi

start a start billentyűnek mind LOAD, mind RUN üzemmódban van feladata. Ha a LOAD/STEP/RUN kapcsoló LOAD állásban van, a start billentyű lenyomásával a programtárban minden információt törölünk, START utasítást írunk be, és megjelöljük az első program kezdetét. STEP állásban a start billentyű minden egyes lenyomására a tárolt program egy-egy lépését végzi el a gép.

Ha a kapcsoló RUN állásban van, a start gomb benyomására a gép megkezdí az első program végrehajtását. Ha a program futása egy HALT kódnál (ld. később) megáll, a start billentyű benyomására a gép a következő HALT-ig, ill. a program végéig folytatja a program végrehajtását. Ha a programnak vége van, a gép mindig az első program START kódjára áll vissza.

skip a skip billentyűnek mind LOAD, mind RUN üzemmódban van szerepe. Ha a LOAD/STEP/RUN kapcsoló LOAD állásban van, a skip és utána a halt billentyű lenyomásával további programok kezdetét jelölhetjük meg. Ha a kapcsoló RUN állásban van, a skip billentyű lenyomásával valamely program kezdetéről, vagy egy HALT pontról a következő program elejére ugorhatunk, és megkezdhetjük ennek végzését. A program ekkor az első HALT-ig ill. a program végéig fut. Ha a gépbe csak egy programot írtunk be, és a program végrehajtása egy HALT-nál megáll, a skip billentyű lenyomására a gép a program hátralevő részét átugorja és ismét az elejéről kezdve hajtja végre. Ez módot ad a főprogramon belüli „programhurok” létrehozására. Ha csak két programot írtunk be, úgy a skip billentyű lenyomására a második, a start billentyű lenyomására az első programot futtatja a gép. Ha kettőnél több programot írtunk be, úgy az első program kivételével minden további program elejére a SKIP utasítással együtt egy HALT utasítást is be kell programozni. A második program elindításához egyszer kell lenyomni a skip billentyűt, a harmadik prog-

ram elvégzéséhez kétszer kell lenyomni a skip billentyűt, az n-edik program elvégzéséhez pedig n-1 alkalommal.

halt a halt billentyűt csak LOAD üzemmódban használjuk. A billentyű lenyomásával egy HALT utasítást programozunk be a program végrehajtásába. Ha RUN üzemben egy HALT utasításhoz ér a program, automatikus végrehajtása megáll és a gép vezérlése visszakérül a használó személyhez. A halt billentyűt általában arra használjuk, hogy a program végrehajtásába szünetet iktassunk be, amikor részeredményt olvashatunk le, továbbá változót írhatunk be a további számításokhoz. A HALT állapotot általában a start billentyű benyomásával szüntethetjük meg, ekkor folytatódik a program végrehajtása, de a skip billentyű benyomásával átugorhatunk a következő vagy a további programokra is.

del a del billentyűt csak LOAD üzemben használjuk. Segítségével a programozásban elkövetett hibát javíthatjuk ki. Akkor kell megnyomni, ha a programtárba bevitt információt törölni akarjuk. A del billentyű lenyomása mindig az utoljára bevitt programlépést törli. A del billentyű második benyomása az utolsó előtti programlépést törli, stb. Ha START vagy SKIP utasítást kísérelünk meg törölni, akkor a del billentyű lenyomására a gép hibajelzést ad. Ha egy SKIP utasítást töröltünk, az azt jelenti, hogy egy egész programot megszüntettünk. A del billentyű ismételt lenyomására eltűnik a hibajelzés, és a törölt SKIP utasítás előtti programlépés is törölődik. Ha a hibajelzés a del billentyű ismételt benyomására sem tűnik el, az azt jelenti, hogy a programtárban minden programot töröltünk, és a programtárban már csak egyetlen START utasítás van. A START utasítás nem törölhető. Ha tévedésből töröltünk egy SKIP utasítást, a skip benyomásával új SKIP utasítást írhatunk be.

Hibakijelzés

A programozható változat programtárolójának kapacitása 100 lépés. Ha több, mint 100 programlépést írunk be, vagy egy SKIP utasítást törölünk, vagy ha egy START kódot kísérelünk meg törölni, hibajelzés jelenik meg az összes tízeses ponttal. Például .1.3.4.5 ... hibajelzést jelent.

Változók bevétele

Ha a LOAD/STEP/RUN kapcsoló RUN állásban van, és a programban HALT utasítás jelenik meg, akkor a használó személy ismét vezérelni tudja a gépet. Ilyenkor lehet változót beírni a további műveletekhez. Változó alatt az összes számbillentyűt (0–9), a CHS valamint a π billentyűt értjük. A változó beírása után megnyomjuk a start billentyűt, és a program a HALT-helytől folytatódik. A halt billentyű lenyomását követő számbefírást nem veszi a gép állandó mennyiségként és így ez nem marad meg a programban.

Állandók bevétele

Ha a LOAD/STEP/RUN kapcsoló LOAD állásban van, úgy minden szám (0–9), a CHS valamint a π állandóként íródik be, ha a beírás előtt nem programoztunk HALT-ot. Az állandó tehát automatikusan íródik be, és a gép a program minden végrehajtásánál állandóként kezeli.

A programozás kezelőszerveinek rendeltetése

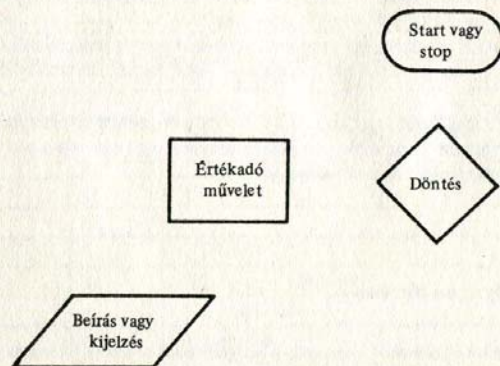
A következő táblázatban összefoglaljuk a program-üzemmód-kapcsoló és a programozó billentyűk rendeltetését.

Billentyű	LOAD-üzemmód	RUN-üzemmód
start	törli a programtárat, és START utasítást ír be	megkezdí az első programot, vagy a HALT-tól folytatja a programot
skip	megelőli az első program után következő programok kezdetét	átugorja a program további részeit, és a következő programot kezdi meg

halt	HALT utasítást programoz be, és ezzel a program majdani végrehajtásakor visszaadja a gép vezérlését a kezelő személynek	nincs rendeltetése
Üzem mód kapcsoló		Rendeltetés
RUN állásban	a start vagy a skip billentyű lenyomására elindul a megfelelő program	
LOAD állásban	a start billentyű lenyomását követően a gép a beírt programot a programtárba viszi és ott tárolja	
átkapcsolás LOAD → RUN	az első program végrehajtásához szükséges átkapcsolás	
RUN → LOAD átkapcsolás	ezt az átkapcsolást akkor végezzük, ha az utoljára beadott programhoz további programlépéseket akarunk bevinni. Ilyenkor a korábbi programozások érintetlenek maradnak. A skip billentyű lenyomásával az eredeti programot többféle programmal is kiegészíthetjük.	
RUN → LOAD, majd ismét → RUN átkap- csolás	az első beadott program végrehajtásához szükséges átkapcsolás. Ez a lehetőség igen hasznos lehet, ha a gép használóját a számolás alatt megzavarják, és elfelejti, hogy a program végrehajtásában hol tartott	
a kapcsoló STEP állásban	a gép a start billentyű minden benyomására a tárolt program egy-egy lépését végzi el	

Program példák

A következő programozási példákkal a programozható számológép sokoldalúságát mutatjuk be. A program menetét a következő programozási jelölésekkel írjuk le:



Ez utóbbinál a beírás változóknak, vagy állandóknak a kezelő által történő beírását, a kijelzés pedig az eredményeknek a gép által történő kiírását jelenti.

A számítógép értelmezési tartományának figyelembevétele a programozás során

Tekintve, hogy a számítógép a túlsordulás jelölésére külön jelet nem használ, így adott program végrehajtása közben a kezelőnek nincs alkalma meggyőződni arról, hogy a program közben kiszámított részeredmények a gép értelmezési tartományán belül vannak-e vagy sem.

Ennek elkerülésére, az olyan számításokat tartalmazó programoknál, ahol egy számítás eredményeként „x” értéke várhatóan a megadott értelmezési tartományon kívül esik, „x” érték kiszámítása után be kell iktatni a programba egy HALT utasítást, így program-futtatás közben ellenőrizhető, hogy a kiszámított „x” érték belül van-e a gép értelmezési tartományán vagy sem.

Ha „x” értéke kívül esik az értelmezési tartományon, akkor értékét módosítani kell (amennyiben ez lehetséges), mert ellenkező esetben a program továbbfuttatása hibás eredményt ad.

Egyszerű programok bevétele

A programozható számítógéppel már a programozás során lehet számolni, azaz a programozás alatt tényleges számértékekkel lehet dolgozni, és megkapjuk a helyes eredményt. Az a tény viszont, hogy a programlépések bevétele tényleges adatok (számértékek) nélkül is történhet, még egyszerű, nem ismétlődő programok esetén is előnyös lehet. Ugyanis a gép e tulajdonsága lehetővé teszi a program kipróbálását oly módon, hogy a programozás befejeztével a programot tényleges adatokkal végrehajtjuk és ellenőrizzük, hogy a kapott eredmény az előre meghatározott eredménnyel megegyezik-e. Ha igen, akkor a programot helyesen vittük be.

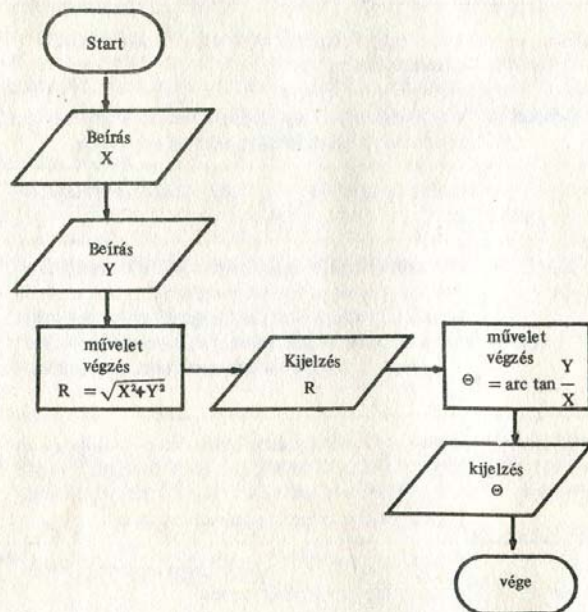
Példa: programozza be a gépet az X és Y derékszögű koordináták R és Θ polárkoordinátákká való átszámítására!

Az $X = 6$ és $Y = 8$ tényleges adattal a következő eredményt várhatjuk:

$$R = \sqrt{X^2 + Y^2} = \sqrt{6^2 + 8^2} = \sqrt{100} = 10$$

$$\Theta = \arctan \frac{Y}{X} = \arctan \frac{8}{6} = \arctan 1.3333333 = 53.12998$$

A kívánt programlefutás:



A feladat beprogramozása

a kapcsoló LOAD állásban

lépés	program	adat	regisztertartalom				M	
			X	Y	Z	T		
1	start							a program kezdetének megjelölése
2	halt							HALT a változó (X) beírásához
		6	6					az X beírása
3	EN		6.	6.				
4	EN		6.	6.	6.			X-et a Z regiszterbe visszük tárolásra
5	X		36.	6.				x^2
6	halt		36.	6.				HALT Y beírásához
		8	8	36.	6.			Y beírása
7	MS		8.	36.	6.		8.	Y-t a memóriában (M regiszter) tároljuk
8	EN		8.	8.	36.	6.	8.	
9	X		64.	36.	6.		8.	Y^2
10	+		100.	6.			8.	$X^2 + Y^2$
11	$\sqrt{\quad}$		10.	6.			8.	$\sqrt{X^2 + Y^2}$
12	halt		10.	6.			8.	HALT az R kijelzéséhez
13	$x \leftrightarrow y$		6.	10.			8.	X-t visszahozzuk
14	MR		8.	6.	10.		8.	Y-t lehívjuk
15	$x \leftrightarrow y$		6.	8.	10.		8.	Y-nak X-szel való osztásához csere
16	$\div$		1.3333333	10.			8.	Y/X
17	arc		1.3333333	10.			8.	
18	tan		53.12998	10.			8.	a gép kijelzi Θ értékét. Vége a programnak

A program felhasználása

Az előbb leírt programozás végrehajtása során bevittük a gépbe a derékszögű koordináták polárkoordinátákká történő átszámításának programját. Számítsuk most át a következő derékszögű koordinátákat polárkoordinátákká:

$$X = 5, \quad Y = 9,$$

$$X = 7, \quad Y = 4,$$

$$X = 3, \quad Y = 3.$$

Állítsuk a LOAD/STEP/RUN kapcsolót RUN állásba.

Beírás	Kijelzés	Megjegyzés
start		megindul az első program, és az első HALT-ig fut
5	5	az első X érték
start	25.	a program tovább fut a következő HALT-ig
9	9	az első Y érték
start	10.29563	a program tovább fut a következő HALT-ig; a gép kijelzi az R értékét
start	60.945697	a program végig lefut; a gép kijelzi az Θ értékét
start	60.945697	ismét kezdődik az első program
7	7	a következő X érték
start	49.	a program fut tovább
4	4	a következő Y érték
start	8.062257	a gép kijelzi a következő R értékét
start	29.744463	a gép kijelzi a következő Θ értékét
start	29.744463	az első program ismét elindul
3	3	a következő X érték
start	9.	a program folytatódik
3	3	a következő Y érték
start	4.24264	a következő R érték kijelzése
start	45.000654	a következő Θ érték kijelzése

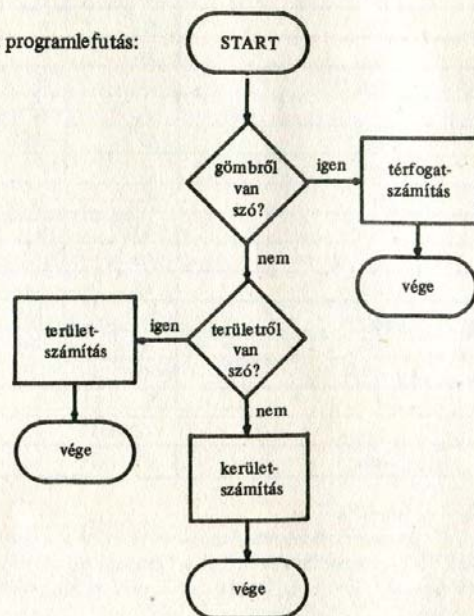
Többszörös programok bevittele

Példa: számítsa ki szükség szerint egy r sugarú kör területét vagy kerületét, vagy egy r sugarú gömb térfogatát!

$$\begin{aligned} \text{a kör területe} &= \pi r^2 \\ \text{a kör kerülete} &= 2 \pi r \\ \text{a gömb térfogata} &= \frac{4}{3} \pi r^3 \end{aligned}$$

A képletekből látható, hogy akármelyiket is használjuk, r mindig szerepel. Így a programot beírhatjuk oly módon, hogy r értékét a program indítása előtt írjuk be a memóriába. Így a programozás során nem kell adatokat beírni.

A kívánt programlefutás:



Programozás, melynél az $r=5$ értéket a program kipróbálására használjuk

lépés	program beírás	adat beírás	regisztertartalom					megjegyzés
			X	Y	Z	T	M	
		5	5					próbaváltó
	MS		5.				5.	tároljuk a program számára
a kapcsoló LOAD-ban								
1	start		5.				5.	az első program kezdetének megjelölése (körterület)
2	π		3.1415927	5.			5.	π bevitele
3	MR		5.	3.1415927	5.		5.	r lehívása
4	EN		5.	5.	3.1415927	5.	5.	
5	x		25.	3.1415927	5.		5.	r^2
6	x		78.539817	5.			5.	πr^2 . Vége az első programnak
7	skip		78.539817	5.			5.	a 2. program kezdetének megjelölése (körterület)
8	halt		78.539817	5.			5.	több mint két programot írunk be, ilyenkor az első program után következő minden programhoz be kell írni egy HALT-ot
9	MR		5.	78.539817	5.		5.	mivel a 8. lépésben szereplő HALT-nál nem kell adatot bevinni, le-nyomhatjuk bármelyik billentyűt az adatbeírókat (0–9, CHS, π) kivéve
10	π		3.1415927	5.	78.539817	5.	5.	π
11	x		15.707963	78.539817	5.		5.	πr
12	2		2	15.707963	78.539817	5.	5.	mivel a számbírás előtt nem volt HALT, a gép a programban konstansként kezeli
13	x		31.415926	78.539817	5.		5.	$2\pi r$. Vége a 2. programnak
14	skip		31.415926	78.539817	5.		5.	a 3. program kezdetének megjelölése (gömbtérfogat)

15	halt	31.415926	78.539817	5.		5.	ugyanaz, mint a 8. lépés
16	MR	5.	31.415926	78.539817	5.	5.	r lehívása
17	3	3.	5.	31.415926	78.539817	5.	
18	y^x	124.99984	0	0	0	5.	r^3
19	π	3.1415927	124.99984			5.	π
20	$\times$	392.69858				5.	πr^3
21	4	4	392.69858			5.	konstans
22	EN	4.	4.	392.69858		5.	
23	3	3.	4.	392.69858		5.	
24	$\div$	1.3333333	392.69858			5.	$\frac{4}{3}$
25	$\times$	523.59809				5.	$\frac{4}{3} \pi r^3$. Vége a 3. programnak
		X	Y	Z	T	M	

A program végrehajtása

Miután beírtuk a gépbe egy r sugarú kör terület- és kerületszámításának, valamint egy r sugarú gömb térfogatszámításának programját, a következő feladatokat is meg tudjuk oldani:

1. feladat: 7 egység sugarú kör területének és kerületének kiszámítása
2. feladat: 9 egység sugarú kör kerületének és 9 egység sugarú gömb térfogatának kiszámítása
3. feladat: 6,5 egység sugarú kör területének és 6,5 egység sugarú gömb térfogatának kiszámítása
4. feladat: 12,35 egység sugarú kör területének és kerületének, valamint 12,35 egység sugarú gömb térfogatának kiszámítása

A LOAD/STEP/RUN kapcsolót RUN állásba kapcsoljuk.

Beírás	Kijelzés	Megjegyzés
7	7	az 1 feladathoz $r = 7$
MS	7.	a sugár értékét tároljuk
start	153.93804	a körterület $r = 7$ esetén
skip	153.93804	az 1 programot átugrani és a 2 programot megkezdeni
start	43.982296	körkerület $r = 7$ esetén
9	9	az új sugár
MS	9.	tároljuk

skip	9.	az 1 programot átugrani, és a 2 programot megkezdeni
start	56.548668	körkerület $r = 9$ esetén
skip	56.548668	az 1 programot átugrani
skip	56.548668	a 2 programot átugrani, és a 3 programot megkezdeni
start	3053.6295	gömbtérfigat $r = 9$ esetén
6.5	6.5	az új sugár
MS	6.5	tároljuk
start	132.73229	körkerület $r = 6.5$ esetén
skip	132.73229	az 1 programot átugrani
skip	132.73229	a 2 programot átugrani
start	1150.3467	gömbtérfigat $r = 6.5$ esetére
12.35	12.35	az új sugár
MS	12.35	tároljuk
start	479.16357	körkerület $r = 12.35$ esetére
skip	479.16357	az 1 programot átugrani
start	77.597338	körkerület $r = 12.35$ esetére
skip	77.597338	az 1 programot átugrani
skip	77.597338	a 2 programot átugrani
start	7890.2338	gömbtérfigat $r = 12.35$ esetére

Iteratív (ismétlődéses) programok bevitale

A programot úgy kell felépíteni, hogy az a végén ismét előlről kezdőd-hessen.

Ily módon ismétlődéses programokat vihetünk be a gépbe. A program is-métlődő részét csak egyszer programozzuk be, a géppel viszont tetszőleges

alkalommal megismételtetjük.

Példa: programozzuk a gépet a következő értékek képlet szerinti négyzet-összegének (X^2) kiszámítására:

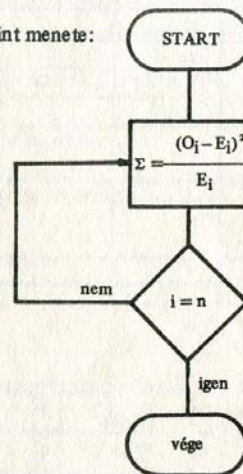
O_i	2	3	5	7	2
E_i	4	4	3	5	3

Az alkalmazandó képlet:

$$X^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i}$$

Ezt a képletet beprogramozhatjuk, és például $n = 5$ alkalommal hajtjuk végre.

A program kívánt menete:



Programozás:

Mivel itt ismétléses programról van szó, a programot nem lehet „igazi” változókkal kipróbálni.

A logikai hibák (pl. 0-val való osztás) elkerülésére használja a program kipróbálásához az 1 „változót”.

az üzemmód kapcsoló LOAD-ban								
lépés	program beírás	adat beírás	regisztertartalom					megjegyzés
			X	Y	Z	T	M	
		1	1					a próbaváltozó (1) bevitele
1	start		1					a program kezdetének megjelölése
2	EN		1.	1.				mivel a program első lépése EN , a program mindig azt az értéket tekinti adatnak, ami éppen a kijelzőn van. Így nincs szükség HALT beprogramozására a változó beviteléhez
3	halt		1.	1.				HALT E _i beviteléhez
4	MS		1.	1.			1.	E _i -t tároljuk az osztáshoz
5	-		0.				1.	O _i – E _i
6	EN		0.	0.			1.	
7	x		0.				1.	(O _i – E _i) ²
8	MR		1.	0.			1.	E _i -t visszahívjuk
9	÷		0.				1.	(O _i – E _i) ² /E _i
10	+		0.				1.	Σ (O _i – E _i) ² /E _i . Vége a programnak
			X	Y	Z	T	M	

A program végrehajtása:

Az X² kiszámítása a következő értékekre:

O _i	2	3	5	7	2
E _i	4	4	3	5	3

A kapcsoló RUN állásban

Beírás	Kijelzés	Megjegyzés
0	0	
EN	0.	

EN	0.	az X és Y regiszter törlése
2	2	O ₁
start	2.	
4	4	E ₁
start	1.	$\sum(O_i - E_i)^2/E_i$ Az alapprogram befejeződött. Hajtsa annyiszor végre, ahányszor kell (n = 5)
3	3	O ₂
start	3.	
4	4	E ₂
start	1.25	összegezve n = 2-vel
5	5	O ₃
start	5.	
3	3	E ₃
start	2.5833333	összegezve n = 3-mal
7	7	O ₄
start	7.	
5	5	E ₄

start	3.3833333	összegezve n = 4-gyel
2	2	O ₅
start	2.	
3	3	
start	3.7166666	összegezve n = 5-tel; vége a program- nak, az X ² értékét kiszámítottuk.

Programhurkok és programelágazások

A start billentyűvel hurkokat, a skip billentyűvel elágazásokat lehet létrehozni a programban. Ez lehetővé teszi olyan ismétléses számítások elvégzését, ahol a kapott eredményeket már rögtön fel is használjuk a számítás folytatásához. Ennek technikáját ismertetjük a következőkben:

Példa: programozza be a gépet a számtani középérték, a szórásnégyzet és a standard eltérés kiszámítására a következő adatokból: 2, 7, 3, 5, 2!

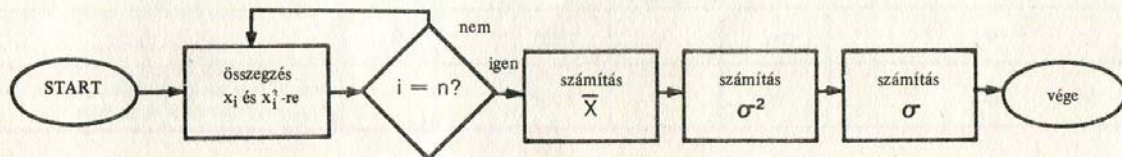
A következő képleteket kell használnunk:

$$\text{számtani középérték} = \bar{x} = \sum x/n$$

$$\text{szórásnégyzet} = \sigma^2 = \frac{\sum x^2 - n(\sum x/n)^2}{n - 1}$$

$$\text{standard eltérés} = \sigma = \sqrt{\sigma^2}$$

A program kívánt menete:



Programozás:

Mivel ismétléses programról van szó, a logikai hibák elkerülésére (pl. nullával való osztás) a 2 látszólagos változót írjuk be a stackbe!

a kapcsoló LOAD állásban							
lépés	program	adat	regisztertartalom				Megjegyzés
	2 EN	EN	X	Y	Z	T	M
			2				
	START						
1	MS		x_i	Σx_i	n	$\Sigma(x)^2$	x_i
2	+		Σx_i	n	$\Sigma(x_i)^2$		Σx_i
3	MR		x_i	Σx_i	n	$\Sigma(x)^2$	x_i
4	$x-y$		Σx_i	x_i	n	$\Sigma(x)^2$	x_i
5	MS		Σx_i	x_i	n	$\Sigma(x)^2$	Σx_i
6	C		x_i	n	$\Sigma(x)^2$		Σx_i
7	roll		n	$\Sigma(x_i)^2$		x_i	Σx_i
8	$x-y$		$\Sigma(x_i)^2$	n		x_i	Σx_i
9	roll		n		x_i	$\Sigma(x_i)^2$	Σx_i
10	roll			x_i	$\Sigma(x_i)^2$	n	Σx_i
11	roll		x_i	$\Sigma(x_i)^2$	n		Σx_i
12	EN		x_i	x_i	$\Sigma(x_i)^2$	n	Σx_i
13	X		x_i^2	$\Sigma(x_i)^2$	n		x_i^2
14	+		$\Sigma(x_i)^2$	n			$\Sigma(x_i)^2$
15	$x-y$		n	$\Sigma(x_i)^2$			
16	1		1	n	$\Sigma(x_i)^2$		Σx_i
17	+		n	$\Sigma(x_i)^2$			Σx_i
18	MR		Σx_i	n	$\Sigma(x_i)^2$		Σx_i
19	skip		Σx_i	n	$\Sigma(x_i)^2$		az $\bar{x}$, σ^2 és σ kiszámítására szolgáló 2 program kezdetének megjelölése

20	$x - y$	n	Σx_i	$\Sigma (x_i)^2$	Σx_i	
21	MS	n	Σx_i	$\Sigma (x_i)^2$	n	
22	$\div$	$\Sigma x_i/n$	$\Sigma (x_i)^2$		n	$\bar{x} = \text{középérték}$
23	halt	$\Sigma x_i/n$	$\Sigma (x_i)^2$		n	HALT $\bar{x}$ értékének kijelzéséhez
24	EN	$\Sigma x_i/n$	$\Sigma x_i/n$	$\Sigma (x_i)^2$	n	
25	X	$(\Sigma x_i/n)^2$	$\Sigma (x_i)^2$		n	
26	MR	n	$(\Sigma x_i/n)^2$	$\Sigma (x_i)^2$	n	
27	X	$n(\Sigma x_i/n)^2$	$\Sigma (x_i)^2$		n	
28	—	$\Sigma (x_i)^2 - n(\Sigma x_i/n)^2$			n	
29	MR	n	$\Sigma (x_i)^2 - n(\Sigma x_i/n)^2$		n	
30	1	1	n	$\Sigma (x_i)^2 - n(\Sigma x_i/n)^2$	n	
31	—	$n - 1$	$\Sigma (x_i)^2 - n(\Sigma x_i/n)^2$			
32	$\div$	$\frac{\Sigma (x_i)^2 - n(\Sigma x_i/n)^2}{n - 1}$			n	szórásnégyzet
33	halt	$\frac{\Sigma (x_i)^2 - n(\Sigma x_i/n)^2}{n - 1}$			n	HALT a σ^2 szórásnégyzet kijelzéséhez
34	$\sqrt{\quad}$	$\sqrt{\frac{\Sigma (x_i)^2 - n(\Sigma x_i/n)^2}{n - 1}}$			n	a σ standard eltérés kijelzése

A program végrehajtása:

Számítsa ki a számtani középértéket, a szórásnégyzetet és a standard eltérést a következő adatokkal: 2, 7, 3, 5, 2!

A kapcsoló RUN állásban:

Beírás	Kijelzés	Megjegyzés
C	0	töröljük a stacket
C	0	töröljük a stacket
C	0	töröljük a stacket
C	0	töröljük a stacket
2	2	x_1

start	2.	az 1 program megkezdődik, program-hurok az x, x^2 és n szummázásához
7	7	x_2
start	9.	
3	3	x_3
start	12.	
5	5	x_4
start	17.	
2	2	x_5

start	19.	az ismétlődéses rész (hurok) befejeződtött; a program most szétágazik azokra a részekre, amelyek az ismétlődéses részben a kapott eredményeket használják fel
skip	3.8	a program megáll az $\bar{x}$ kijelzésére
start	4.7	a program megáll a σ^2 kijelzésére
start	2.1679483	a program vége, a gép kijelzi a σ értéket.

A programozás menetének összefoglalása

1. A programozás megkezdése előtt töröljük a kijelzést. Ehhez nyomjuk le a C billentyűt annyiszor, amennyi szükséges ahhoz, hogy a kijelzőn nulla jelenjen meg. A memória törlésére nyomjuk le a 0 és MS billentyűt.
2. A programozáshoz kapcsoljuk a LOAD/STEP/RUN kapcsolót LOAD állásba.
3. Jelöljük meg az első program kezdetének helyét a start benyomásával. Az összes következő program kezdetét a skip benyomásával jelöljük meg. Ha több mint két programot írunk a gépbe, úgy az első kivételével mindegyikhez egy HALT utasítást kell beprogramozni ahhoz, hogy az illető programhoz hozzá tudjunk férni.
4. Ha meg akarjuk szakítani a programot változó beírása, vagy az eredmény kijelzése céljából, a kívánt helyen nyomjuk le a halt billentyűt.
5. Állandó a kívánt szám bebillentyűzésével vihető be. Az így bevitt állandó a program részévé válik.
6. Változó beírására nyomja meg a halt billentyűt, majd írja be a kívánt számot. Ez a szám nem fog a program részévé válni, viszont kipróbálhatjuk vele a programot.

7. Ha a programot végre akarjuk hajtani, a LOAD/STEP/RUN kapcsolót RUN állásba kell kapcsolni.
8. Az 1. program indítása a start billentyűvel történik, a második program indítása a skip benyomásával, az n-edik program pedig a skip n-1-szeri benyomásával.

A KEZELÉSI ÚTMUTATÓBAN TALÁLHATÓ JELÖLESEK ÉS Néhány kifejezés magyarázata

start	indítás
halt	megállás
skip	ugrás
del	delete – törlés. A legutóbb beírt programlépést törli
load	töltés – a program beírása
step	léptetés – a műveletek lépésenkénti végrehajtása
run	futás – a programba beírt műveletek gyors lefuttatása

Műszaki szerkesztő: Arday Janka Zsolt
Felelős szerkesztő: Réti Sándor
Nyomdai szerkesztő: Szöllőssy Csaba
Felelős kiadó: Dóka László

Foto Optika ISz. Bp.
Felelős vezető: Halász Lajos



Gyártja: HIRADÁSTECHNIKA
1400 Budapest, Pf. 23
Telex: 22-6151 HTSZH