

*Az előző részben kitűzött feladatok megoldása*

**Feladat.** a) Készítsünk SUBROUTINE-t, mely egy háromszög csúcspontjainak koordinátáiból kiszámítja a területet és kerületet, a súlyvonalak hosszát és a súlypont koordinátáit, a háromszög köré írható kör középpontjának koordinátáit és a kör sugarának hosszát.

b) Írjunk programot, amelyben a csúcspontokat értékadó utasításokkal határozzuk meg, majd felhasználva az elkészített SUBROUTINE-okat, kinyomtatja az összes számított eredményt, megfelelő feliratokkal.

**Megoldás.** Az alábbiakban a MASTER szegmenst és a azubrutin szegmenseket egyben, egy komplett programban mutatjuk be. A program utolsó szegmense után a FINISH utasítás a teljes program záró utasítása. A hatodik pozíción (a két „függőleges” vonal között) elhelyezkedő karakter azt jelenti, hogy az utána következő utasításmező folytatása az előző sorbeli utasításmezőnek. Az ilyen sorokat folytató sornak nevezzük. 20-nál több sor nem kapcsolható ily módon össze. A program egy lehetséges alakja:

	MASTER PR11
	DIMENSION PONT(3,2), SUP(2), SUV(3), CENT(2)
	PONT(1,1) = -3
	PONT(1,2) = 1
	PONT(2,1) = 4
	PONT(2,2) = 6
	PONT(3,1) = 8
	PONT(3,2) = -5
	CALL TERKER(PONT, ATER, AKER)
	CALL SULLY(PONT, SUP, SUV)
	CALL KOER(PONT, CENT, RADIUS)
	WRITE(3,5)((PONT(I,J), J = 1,2)J = 1,3)ATER, AKER, (SUP(K)
5	Z K = 1,2)(SUV(N), N = 1,3), (CENT(N), N = 1,2), RADIUS)
	FORMAT(1H1,5(/),40X,27HAAROMSZOEGVCSUCSPONTJAI : //3(40X
X	,2HX =,F10.3,3X,2HY =,F10.3)///40X,20HAHAAAROMSZOEGVTERUEL,
Y	4HETE :,F15.5//40X,24HAHAAAROMSZOEGVKERUELETE :,f15.5/////
Z	40X,25HAVSULYPONTVKOORDINAATAI :,5HVVVX =,F15.5,5HVVVY =,
W	F15.5//40X,21HAVSULYVONALAKVHOSSZA :,3(F15.5,5X)////40X,
V	36HAVKOEREEVIRHATOOVKOERVKOEZEPPONTJA :,5HVVVX =,
P	F15.5,5X,
U	SHVVVY =,F15.5//40X,29HAVKOEREEVIRHATOOVKOERVUGARA :
Q	,F15.5)
	STOP
	END
	SUBROUTINE TERKER(P, TER, KER)
	REALKER, P(3,2)
X	TER = 0.5(*P(1,1)*P(2,2) + P(1,2)*P(3,1) + P(2,1)*P(3,2) -
	P(3,1)*P(2,2) - P(3,2)*P(1,1) - P(2,1)*P(1,2))
	KER = 0
	DO 7 I = 1,3
	J = I + 1
	IF(4 - J)0,0,7
	J = 1
	KER = KER + TAAV(PI,1), P(J,1), P(I,2), P(J,2))
	RETURN
	END
	SUBROUTINE SULLY(P, SP, SV)
	DIMENSION P(3,2), SP(2), SV(3), FX(3), FY(3)
	DO 5 I = 1,2
	SP(I) = (P(1,1) + P(2,1) + P(3,1))/3.
	DO 4 J = 1,3
	K = J + 1
	IF(4 - K)0,0,3

```

      K = 1
      L = K + 1
      IF(4 - L)0,0,2
      L = 1
      FX(J) = (P(K, 1) + P(L, 1))/2.
      FY(J) = (P(K, 2) + P(L, 2))/2.
      DO 6 I = 1,3
      SV(I) = TAAV(FX(I), P(I, 1), FY(I), P(I, 2))
      RETURN
      END
      SUBROUTINE KOER(P, CK, RO)
      DIMENSION P(3, 2), CK(2), RM(2), A(29, FX(2), FY(2))
      DO 1 I = 1,2
1      RM(I) = -(P(I + 1, 1) - P(1, 1))/(P(I + 1, 2) - P(1, 2))
      DO 2 I = 1,2
      FX(I) = (P(1, 1) + P(I + 1, 1))/2.
2      FY(I) = (P(1, 2) + P(I + 1, 2))/2.
      DO 3 I = 1,2
3      A(I) = FY(I) - RM(I) * FX(I)
      CK(1) = -(A(1) - A(2))/(RM(1) - RM(2))
      CK(2) = (RM(2) * A(1) - RM(1) * A(2))/(RM(2) - RM(1))
      RO = TAAV(CK(1), P(1, 1), CK(2), P(1, 2))
      RETURN
      END
      FUNCTION TAAV(X1, X2, Y1, Y2)
      TAAV = SQRT((X1 - X2) * *2 + (Y1 - Y2) * *2)
      RETURN
      END
      FINISH

```

A szubrutinok algoritmusai közismert koordináta geometriai eljárások, melyeket itt nem részletezünk.

## 5. Olvasztás lyukkártyáról

### 5.1 Az adatlap

Rovatunkban kizárólag számok olvasztásával foglalkozunk, amelyeket ún. adatlapra írva adunk le lyukasztásra. Mivel adatok beolvasása a lyukkártya valamennyi pozíciójairól egyformán lehetséges, ezért az nem osztható eleve meghatározott mezőkre. A könnyebb áttekinthetőség érdekében IO egyforma, 8 karakter széles sávot szoktunk behúzni az adatlapra, amely a programlaphoz hasonlóan négyzethálós papírra helyezendő el. A fejléc is megegyezik a korábbival.

Néhány fontos tudnivaló:

- az adatlapon kizárólag a tíz számjegy és a mínusz jel karaktere jelenhet meg;
- az adatbevitelnél nincs helyközérzékenység, az összetartozó számjegyek, ill. számjegy és mínusz jel között betűköz nem hagyható;
- az adatlapra írt minden előjel nélküli szám pozitív;
- két adat közé semmilyen „elválasztó jelet” nem írunk az adatlapon.

### 5.2. Az olvasztó utasítás

A kártya olvasztás utasítása a READ (jelentése: olvasd, ejtsd: rid). Ezt követi zárójelpár között a logikai perifériaszám és egy FORMAT utasítás címkéjével azonos szám. A zárójel után áll az ún. READ-lista, vagyis olyan azonosítók sorozata, amelyek a beolvasott értékeket felveszik. Az egyszerű listájú READ utasításra példa:

```

| | READ(1, 17)A, B(5), IND, K7, K9, K13

```

Rovatunkban a kártynolvasó logikai perifériaszáma mindig 1 lesz. Az olvasó utasításhoz is kötelezően tartozik egy FORMAT utasítás, melyre címkével hivatkozunk. Ez írja elő a kártyáról beolvasott számnadat formáját, típusát, nagyságrendjét. Emiatt a következő összetartozó dolgokat kell figyelemmel kísérnünk:

- u kártya valamely részének tartalmát,
- a FORMAT erre vonatkozó specifikációját,
- a READ-lista megfelelő elemét és
- a beolvasott szám értékét.

A FORMAT után zárójelben állnak azok a specifikácók, amelyek a kártya tartalmának bevitelére vonatkoznak. Rovatunkban az olvasztás I, F és X specifikációval foglalkozunk, de ezeknek sem tárgyaljuk valamennyi lehetséges felhasználását. A példának hozott READ utasításhoz az alábbi FORMAT tartozhat:

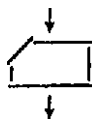
```

17 | | FORMAT(F8.3, F10.5, 10X, 16, 3I10)

```

Néhány szabály:

- Ha a READ-lista egyszerű, akkor egy READ utasítás egy kártya beolvastatását váltja ki, ezért minden ilyen utasításhoz pontosan egy adatkártyának kell tartoznia.
- A kártyán legalább annyi adatnak kell állnia, amennyi a READ-lista elemeinek száma.
- A FORMAT-lista specifikációi rendre és típusra a rá hivatkozó READ utasítás listájában szereplő azonosítókra kell hogy vonatkozzanak.
- Az I specifikáció egész típusú, az F specifikáció valós típusú számként olvastatja be a kártya tartalmát.
- Az lw ill. Fw.d alakú specifikációkban a w mezőszélesség a kártyáról beolvasott karakterek számát jelenti. A specifikációk által meghatározott mezők a kártya első karakterpozíciójánál kezdődnek és a specifikációk sorrendjében következnek egymás után, legfeljebb a kártya végéig.
- Az nX specifikáció ismétlési tényezője azt írja elő, hogy a kártya hány karakterét hagyja figyelmen kívül az olvasás.
- A fentiekből következik, hogy az egyszerű READ-listához tartozó FORMAT-listában a specifikációk által „lekötött” pozíciók számának összege nem lehet 80-nál több.
- Az X specifikációval kihagyott, valamint a FORMAT-lista által le nem kötött (a kártya végére eső) kártyapozíciók tartalma tetszőleges lehet, nem olvasódik a tárba.
- A programban egymás után végrehajtásra kerülő egyszerű listájú READ utasítások egymás után következő adatkártyákat olvasnak be. A programozó feladata, hogy az. adatlapon ennek a sorrendnek megfelelően következzenek soronként a beolvasandó, egy-egy kártyára lyukasztandó adatok.
- Blokkdiagramban a beolvasást ilyen blokkal jelezzük:



5.2.1 Az I specifikációjával való olvastatásra tekintsük a következő példát:

6	READ(1, 6)IP, JA, KS FORMAT(I10, I8, 12X, I5)
---	--

A specifikációk a kártyán négy egymást követő mezőt határoznak meg. A kártya 1–10 karaktereit az I10, a 11–18 karaktereit az I8 specifikáció olvassa be. Ezután 12 karakterpozíció tartalma figyelmen kívül marad és a 31–35 karakterpozíciók tartalmát, az I5 specifikációval olvastatjuk be. Vegyük pl. az első, I10-el beolvasandó mezőt. Tegyük fel, hogy erre a –136 számot lyukasztjuk. Ezt többféleképpen tehetjük, és ennek megfelelően különböző nagyságrendű számok kerülnek beolvasásra. Ezt az alábbi táblázat mutatja.

a kártyára lyukasztva:	tárba olvasva I10-el:
▽▽▽▽▽▽ – 136	–136
▽▽▽▽▽ – 1367	–1360
▽▽▽ – 136▽▽▽	–136000
▽▽ – 136▽▽▽▽	–1360000

Látnivaló, hogy

- a lyukasztott szám csakis akkor kerül eredeti nagyságrendjében a tárba, ha a mezőre *jobbra tömörítetten* írjuk be;
- a mezőben a számtól jobbra álló betűközök mint zérus számok olvasódnak be a tárba;
- a szám bal felé való elcsúsztatása a mezőben csak addig engedhető meg, amíg a zérusokkal kiegészített szám a gépi számábrázolás korlátait meg nem haladja.

Ha például a hozott utasítás olyan kártyát olvastat be, melynek első 35 karaktere az alábbi:

▽▽▽ – 136000▽▽▽5963427▽▽▽▽17683215075

akkor a READ-lista azonosítói közül IP-nek 8212; 136000, JA-nak 59634, KS-nek 15075 lesz az értéke.

5.2.2. Az F specifikáció megismeréséhez tekintsük a következő példát:

4	READ(1, 4)ALFA, BETA, EPS FORMAT(5X, F6.3, F10.5, F12.7)
---	---

A READ – listahromazonostjhozaFORMAT-lista három specifikációt tartalmaz. A negyedik specifikáció (sorrendben az első) az első 5 karakterpozíció figyelmen kívül hagyását írja elő. Az F specifikáció a hozzá tartozó mezőt két részre osztja. A mező jobb szélén annyi karaktert különít el, amennyit a tizedes törtjegyek száma, a d előír. Ezen karakterek tartalma képezi a beolvasott szám tört részének jegyeit. A mező bal részében álló karakterek adják meg a beolvasott szárn egész részének számjegyeit. Ha F.10.5 specifikációval olvastatunk be,

és a kártya tartalma	akkor a beolvasott szám értéke
▽▽▽2369▽▽▽	23,69000
▽▽▽ – 2369▽▽	–2,36900
▽▽▽▽▽▽2369	0,02369
2369▽▽▽▽▽▽	23690,000000

A példából leolvasható, hogy

- a jobb oldalon álló betűkzók zérus számként kerülnek a tárba,
- bármilyen F specifikációval beolvasott szám eggyel kevesebb karakterből áll, az ugyanezen szám nyomtatásánál felhasznált karakterek számánál, mivel utóbbiban a tizedes pont is megjelenik.

Ha például hozott utasítás olyan kártyát olvastat be, amelynek első 33 karaktere az alábbi:

▽53▽▽231709▽▽ – 6207▽▽▽▽▽▽▽▽▽▽33305

akkor ALFA értéke 231,709, BETA értéke –62,07000 és EPS értéke 0,0033305 lesz.

Következő rovatunkban az olvastató utasítás használatának bemutatását befejezzük. Utána különböző feladatokon alkalmazni fogjuk eddig tanult ismereteinket. Azok részére, akik az elmúlt oktatási évben nem tudták rendszeresen olvasni rovatunkat, *összefoglaló jegyzetet készítettünk a FORTRAN programozási nyelv alapjairól*. Kérjük, hogy aki a jegyzetre igényt tart, az küldjön a szerkesztőség címére egy postai levelezőlapot az alábbi szöveggel:

Megrendelem a „FORTRAN programozási nyelv alapjai” c. jegyzetet.

Olvasható aláírás és pontos lakcím.

Címünk: Kömal Szerkesztőség, 1443 Budapest, Postafiók 129.

Sajnálattal közöljük, hogy Gergely János kivált a rovat szerkesztéséből. (Emiatt, a levelek megválaszolásában elmaradás történt.) Kérjük olvasóinkat a továbbiakban a leveleket és feladatmegoldásaikat az alábbi címre küldjék:

Dr. Ada-Winter Péter  
Munkaügyi Minisztérium  
Számítástechnikai Intézet  
1089 Budapest  
Reguly A. u. 57–59.