A legutóbb kitűzött feladatok és azok megoldása:

1. feladat. Készítsünk FUNCTION szubrutint, amely tetszőlegesen megadott koordinátájú két pont közötti távolság értékét számítja ki.

Megoldás. Mivel a távolságot a $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$ képlettel számítjuk, tulajdonképpen utasításfüggvénnyel is megoldható lenne a feladat. FUNCTION szubrutinnal a programrész az alábbi:

```
      FUNCTION TAAV X1,X2,Y1,Y2
      TAAV=SQRT((X1–X2)**2+(Y1–Y2)**2)
      RETURN
      END
```

2. feladat. Készítsünk még egy FUNCTION szubrutint, amely legfeljebb 30 számadat átlagát számítja ki (az adatokat egy 30 elemű tömb, az adatok számát egy egész típusú azonosító tartalmazza).

Megoldás. Az átadásra kerülő adatokat egy tömbazonosító tartalmazza, ezért ez a feladat utasításfüggvénnyel már nem oldható meg.

Az adatok pontos számát előre nem tudjuk, emiatt azt az N számot is meg kell adnunk, amely a tömbben levő átlagolandó adatok számosságát jelenti. A feladat értelmében természetesen a $0 < N \leqq 30$ feltételnek kell teljesülnie.

N-re a szubrutinban vizsgálatot tartunk és ha az a korlátokon kívülre esik, akkor az átlagértéknek számítás nélkül zérus értéket adunk, és ezt szövegben is kinyomtatjuk.

Az átlag számításánál az N-nel való osztáshoz a FLOAT standard utasításfüggvényt használjuk, hogy az egész típusú számból valós típusú legyen. Pl. ha $N = 5$, akkor az X=FLOAT(N) hatására X = 5,0 értékű lesz.

A szubrutin egy lehetséges formája az alábbi:

```
      FUNCTION AATL(ADAT,N)
      DIMENSION ADAT(30)
      AATL=O.
      IF(N)1,1,0
      IF(N–30)0,0,4
      I=1
3     AATL=AATL+ADAT(I)
      IF(N–I)2,2,0
      I=I+1
      GO TO 3
2     AATL=AATL/FLOAT(N)
      RETURN
4     WRITE(3,6)
6     FORMAT(//10X,18HN EERTEEKE TUL NAGY)
      RETURN
1     WRITE(3,5)
5     FORMAT(//10X,23HNINCS MIBOEL AATLAGOLNI)
      RETURN
      END
```

3. feladat. Készítsünk programot, melyben öt koordinátáival adott pont összes lehetséges távolságát és ezek átlagait számítjuk ki a fenti két szubrutin segítségével.

Kinyomtatandók feliratozva az adott öt pont koordinátái, ugyancsak feliratozva a pontok közötti lehetséges távolságok és végül újabb felirat mellett a távolságok átlaga.

Megoldás. Nagyon fontos megjegyzések:

– ha a szubrutin a hívóprogramtól tömböt vesz át, akkor az átadandó tömbnek a hívóprogramrészben is deklaráltnak kell lennie;

– a hívóprogramban deklarált tömb mérete nem lehet kisebb, mint a szubrutin által átvett tömbé.

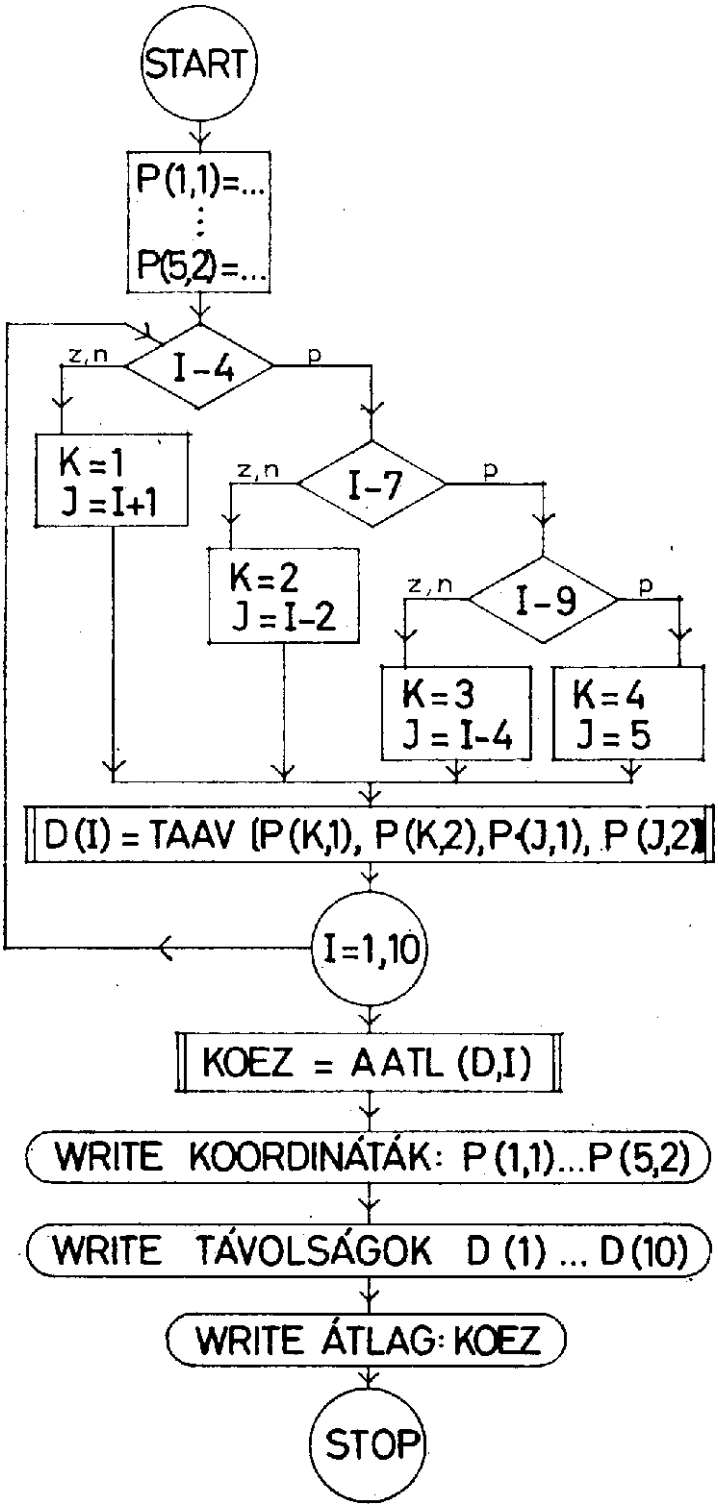
Jelen esetben pl. csak tíz adat átlagát kívánjuk számítani (mivel öt pontnak tíz távolsága lehetséges), a **MASTER** programrészben mégis 30 elemű vektort kell deklarálnunk, mivel a szubrutin ilyen méretű tömböt „vár el”.

A távolságokat négy csoportban számítjuk: először a P_1 pont távolságát vesszük a másik négy ponttól, azután a P_2-ét a többi, nála nagyobb indexű három ponttól, majd a P_2 pont távolságát P_4 és P_5-től, végül a P_4 távolságát P_5-től.

A távolságokat a D(I) tömbbe helyezzük el éspedig

$1 \leqq 1 \leqq 4$ esetén az első csoport,
$5 \leqq 1 \leqq 7$ esetén a második csoport,
$8 \leqq 1 \leqq 9$ esetén a harmadik csoport és
$1 \leqq 10$ esetén a negyedik csoport

távolságait.



A programban a csoport számozását a K azonosító jelenti. A P_1, P_2 stb. pontok koordinátáit programunkban egyszerű értékadással határozzuk meg és egy 5-ször 2-es méretű tömbbe töltjük be. J a második pont koordinátáihoz tartozó index. A program blokkdiagramja az 1. ábrán, és egy lehetséges megírása az alábbiakban látható.

```
      MASTER PKIL1
      REAL KOEZ
      DIMENSION P(5,2),D(30)
      P(1,1)=5.1
      P(1,2)=3.7
      P(2,1)=-1.2
      P(2,2)=6.3
      P(3,1)=-7.5
      P(3,2)=3.4
      P(4,1)=-4.6
      P(4,2)=-4.4
      P(5,1)2.1
      P(5,2)=-3.9
      DO 1 I=1,10
      IF(I–4)0,0,2
      K=1
      J=I+1
      GO TO 1
    2 IF(I–7)0,0,3
      K=2
      J=I–2
      GO TO 1
    3 IF(I-9)0,0,4
      K=3
      J=I–4
      GO TO 1
    4 K=4
      J=5
    1 D(I)=TAAV(P(K,1),P(K,2),P(J,1),P(J,2))
      KOEZ=AATL(D,I)
      WRITE(3,5)
    5 FORMAT(1H1////20X,14HKOORDINAATAAK:/)
      WRiTE(3,6)((I,P(1,1),I,P(I,2)),I=1,5)
    6 FORMAT(5(22X,1HX,I1,1H=,F6.3,3X,1HY,I1,1H=,F6.6,/))
      WRITE(3,7)
    7 FORMAT(///20X,13HTAAVOLSAAGOK:)
      WRITE(3,8)((I,D(I)),I=1,10)
    8 FORMAT(10(/22X,2HD(,12,3H)=,F10.3))
      WRITE(3,9)KOEZ
    9 FORMAT(///20X,21HTAAVOLSAAGOK AATLAGA:,F10.3)
      STOP
      END
```

4.3 A SUBBROUTINE nevű szubrutin

A FUNCTION-hoz abban hasonlít, hogy
– az őt hívó programrésztől független programrész;
– egy azonosítója van, mely után zárójelben formális paraméterek állnak;
– ugyanúgy megkülönböztetünk benne logikai végződést, amelyből többet is tartalmazhat (RETURN) és fizikai végződést, melyből mindig pontosan egy van (END);
– a benne szereplő tömböket, illetőleg azonosítókat az előírt módon külön deklarálni kell.

Lényeges különbség a kétfajta szubrutin között, hogy
– a SUBROUTINE-t a hívóprogramban külön utasítás hívja be, melyről a legközelebbi rovatunkban lesz szó;
– a SUBROUTINE azonosítója nem vesz fel számértéket, a kiszámított eredményt, illetőleg eredményeket az aktuális paraméterek tartalmazzák.

Az alábbi programrész első sora egy szubrutin nyitó utasítása:

```
      SUBROUTINE TRIANG (PONT, OLD, TER)
      DIMENSION PONT(3,2),OLD(3)
```

A zárójelben álló azonosítókról nem látható, hogy azok tömb vagy szám azonosítói. A második sorban levő deklaráció nyújt erre információt.

A zárójelben álló azonosítók közül nem mindegyiknek kell értéket adnunk a szubrutin hívása előtt. Elképzelhető például, hogy a PONT három soros két oszlopos mátrix, amely egy háromszög csúcspontjainak koordinátáit tartalmazza és aktuális paraméterként ezt adjuk át a szubrutinnak. Ezekből az kiszámítja az oldalak hosszát, amelyeket az OLD azonosítójú három komponensű vektorban helyez el, a TER azonosítónak pedig a terület mérőszámát adja. Ebben az esetben az OLD és a TER csak a szubrutin végrehajtása után rendelkezik a kívánt aktuális értékekkel. A hívóprogram ezeket fogja átvenni.

Példa. Készítsünk szubrutint, amely a háromszög csúcspontjainak koordinátáiból kiszámítja a magasságvonalak hosszát és a magasságpont koordinátáit.

A háromszögben csak abban az esetben érdemes a számítást elvégezni, ha a három pont nem esik egy egyenesbe, azaz ha a

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

determináns értéke nem zérus. (Scharnitzky: Mátrix számítás 76. old.) Ezt megvizsgáljuk, és ha zérus, akkor a kimenő adatokat nem számítjuk, hanem az N egész típusú azonosítónak 1 értéket adunk. Ellenkező esetben $N = 0$ lesz, és a számítást elvégezzük. Bevezetjük az

$$x_i - x_j = u_j \qquad \text{és} \qquad y_i - y_j = v_j$$

jelöléseket, ahol $i = 1, 2, 3$ esetén $j = 2, 3, 1$ értékeket veszi fel rendre.

A magasságvonalak talppontjainak $Ti(\widetilde{x}i, \widetilde{y}i)$ koordinátáit az alábbi képletek adják:

$$\widetilde{x}_i = \frac{u_j \cdot v_j}{u_j^2 + v_j^2} \left(\frac{v_j}{u_j} \cdot x_j + \frac{u_j}{v_j} \cdot x_i + v_i \right)$$
$$\widetilde{y}_i = \frac{u_j \cdot v_j}{u_j^2 + v_j^2} \left(\frac{u_j}{v_j} \cdot y_j + \frac{v_j}{u_j} \cdot y_i + u_i \right)$$

ahol $i = 1, 2, 3$ értékek mellett $j = 2, 3, 1$.

Ezek a képletek azonban csakis akkor használhatók, ha $u_j \neq 0$ és $v_j \neq 0$.

$u_j = 0$ esetében $\widetilde{x}_i = x_j$, $\widetilde{y}_i = y_i$; $v_j = 0$ esetében pedig $\widetilde{x}_i = x_i$ és $\widetilde{y}_i = y_j$.

Ezek után a P_i pontból húzott magasságvonal hossza ($i = 1, 2, 3$):

$$D_i = \sqrt{(x_i - \widetilde{x}_i)^2 + (y_i - \widetilde{y}_i)^2}.$$

Az $M(x, y)$ magasságpont koordinátáit az alábbi képletek adják:

$$x = \frac{u_j \cdot v_k \cdot x_i - u_k \cdot v_j \cdot x_j + v_i \cdot v_j \cdot v_k}{u_j \cdot v_k - u_k \cdot v_j}$$
$$y = \frac{u_k \cdot v_j \cdot y_i - u_j \cdot v_k \cdot y_j + u_i \cdot u_j \cdot u_k}{u_k \cdot v_j - u_j \cdot v_k}$$

ahol az indexek $i = 1, 2, 3$ mellett $j = 2, 3, 1$ és $k = 3, 1, 2$ értékeket vehetik fel rendre.

Természetesen elegendő i, j és k egyféle számhármasával számolnunk.

Szubrutinunk a hívóprogramtól a CS azonosítójú háromsoros, kétoszlopos tömböt veszi át. Ez soronként tartalmazza a háromszög csúcspontjainak x és y koordinátáit. A hívóprogramban gondoskodunk arról, hogy az egymás alatti sorokban a koordináták a háromszög csúcspontjainak bal irányú körüljárása szerint következzenek. A többi azonosító értéke (VON, MP, N) a hívás pillanatában érdektelen, mivel a SUBROUTINE MAGAS fog értéket adni nekik.

Mivel a MP tömb automatikusan egész típusú lenne, azért azt külön REAL-lal deklaráltuk. Szubrutinunkban további, az aktuális paraméterek közt nem szereplő tömbökre is szükség van, melyeket ugyancsak deklaráltunk. A tömbök értelme:

VON a magasságvonalak hosszát, MP a magasságpont koordinátáit tartalmazza. U és V a korábbi számításban már bevezetett u és v változóknak felelnek meg, XH és YH a talppontok koordinátáit jelölik.

A zérusra való vizsgálat ebben a programunkban is zérus bizonyos környezetébe eső számokra történő vizsgálatot jelent.

A szubrutin egy lehetséges megírása az alábbi:

```
      SUBROUTINE MAGAS(CS,VON,MP,N)
      DIMENSION CS(3,2), VON(3), U(3), V(3), XH(3), YH(3)
      REAL MP(2)
      DET=CS(1,1)*Cs(2,2)+CS(1,2)*CS(3,1)+CS(2,1)*CS(3,2)
      DET=DET-CS(3,1)*CS(2,2)-CS(3,2)*CS(1,1)-CS(2,1)*CS(1,2)
      IF(ABS(DET-10.**(-6))0,0,1
      N=1
      RETURN
    1 N=0
      DO 2 I=1,3
      IF(I-3)0,3,3
      U(I)=CS(I,1)-CS(I+1,1)
      V(I)=CS(I,2)-CS(I+1,2)
      GO TO 2
    3 U(I)=CS(I,1)-CS(I-2,1)
      V(I)=CS(I,2)-CS(I-2,2)
    2 CONTINUE
      DO 4 1=1,3
      IF(I-2)0,8,9
      J=2
      GO TO 7
    8 J=3
      GO TO 7
    9 J=1
    7 IF(ABS U(J))-0.001)11,11,0
      IF(ABS(V(J))-0.001)10,10,0
      Z=U(J)*V(J)/(U(J)**2+V(J)**2)
      XH(I)=Z*(V(J)*CS(J,1)/U(J)+U(J)*CS(I,1)/V(J)+V(I))
      YH(I)Z*(U(J)*CS(J,2)/V(J)+V(J)*CS(I,2)/U(J)+U(I))
      GO TO 4
   11 XH(I)=CS(J,1)
      YH(I)=CS(I,2)
      GO TO 4
   10 XH(I)=CS(I,1)
      YH(I)=CS(J,2)
    4 VON(I)=SQRT((CS(1,1)-XH(I))**2+(CS(1,2)-YH(I))**2)
      MP(1)=U(2)*V(3)*CS(1,1)-U(3)*V(2)*CS(2,1)+V(1)*V(2)*V(3)
      MP(1)=MP(1)/(U(2)*V(3)-U(3)*V(2))
      MP(2)=U(3)*V(2)*CS(1,2)-U(2)*V(3)*CS(2,2)+U(1)*U(2)*U(3)
      MP(2)=MP(2)/(U(3)*V(2)-U(2)*V(3))
      RETURN
      END
```

Feladat

Szubrutin készítendő, amely tetszőleges konvex négyszög csúcspontjainak koordinátáiból kiszámítja annak területét és kerületét. A bal körüljárás szerint következő csúcspontok koordinátáit egy mátrix egymás alatti sorai tartalmazzák. A szubrutin megfelelő értelmű szöveg nyomtatásával jelezze, ha a négy pont vagy közülük bármelyik három egy egyenesbe esik. Ilyen esetekben a területet, kerületet ne számolja.