

Feladatmegoldások

1. Azt a korábbi feladatot, melyben egy sorba kellett kinyomtatni a természetes számokat, ismételjük meg úgy, hogy a számok 91-től növekedjenek egyesével és a program **WRITE** utasításba épített ciklussal készüljön.

Megoldása az alábbi program lehet:

```
2  MASTER POET
   WRITE (3,2)(K, K = 91, 124)
   FORMAT (/913,2514)
   STOP
   END
```

Érdekesége, hogy az egy sorba nyomtatandó számokat is maga a **WRITE**-ban helyezett ciklus képezi. Emiatt nem szerepel számtömb és annak deklarációja.

2. Készítsünk programot, amely a $2x^3 - 4y^2 + 6x^2 - 8xy + 10y - 12$ polinomnak három tizedesjegy pontossággal kinyomtatja a értéktáblázatát a $-1 \leq x \leq +1$ és $-12 \leq y \leq +12$ határok között, $\Delta x = 0,2$ és $\Delta y = 0,5$ lépésközök mellett.

Az adott polinomot most is célszerű úgy felbontani, ahogy azt a legutóbb a **PHAR** nevű programunkban tettük:

$$2x^3 - 4y^2 + 6x^2 - 8xy + 10y - 12 = x(x(2x + 6) - 8y) + y(10 - 4y) - 12.$$

Látnivaló, hogy míg a bal oldalon (a hatványokat szorzatoknak tekintve) összesen 20 műveletet kell végezni, addig a jobb oldalon (a zárójel szerinti műveleti sorrendet betartva) csak 13 művelet végzendő. Emiatt a jobb oldali kifejezés helyettesítési értékét gép rövidebb idő alatt számítja. Polinomok ilyen típusú azonos átalakítást **Horner-féle elrendezésnek** szoktuk nevezni.

Ezek után a program így írható:

```
2  MASTER PNEGY1
   DIMENSION Z(49,11)
   Y = -12.0
   DO 1 I=1,49
     X = -1.0
     DO 2 J=1,11
       Z(I,J)=X*(X*(2*X+6)-8*Y)+Y*(10-4*y)-12
       X = X + 0.2
     1   Y = Y + 0.5
   WRITE (3,9)((Z(I,J), J=1,11), I=1,49)
   9   FORMAT (1H1//49(5X,11F11,3//))
   STOP
   END
```

A nyomtató utasítást úgy adtuk meg, hogy a mátrixnak 49 sora és 11 oszlopa legyen. Az egy sorba kerülő számok mezőszélessége 11. Ezen belül az egész rész (figyelembe véve egy space-t, előjelet, tizedespontot) legfeljebb 5 jegyből állhat.

Néhány próbaszámítás árán meggyőződhetünk arról, hogy várható helyettesítési értékeink részére ez a mezőszélesség elegendő.

2/b A fenti feladatot bővítsük úgy, hogy nyomtatni kell még annak a polinomnak a helyettesítési értéktáblázatát is, melyet az eredetileg adott polinomból úgy nyerünk, hogy valamennyi együttható értékét egyidejűleg 1-gyel csökkentjük. Ezután nyomtassa ki a program azt a mátrixot is, mely megmutatja, hogy a csökkentett együtthatóval képzett helyettesítési értékek az eredetieknek hány százaléka. Vegyük figyelembe, hogy az eredeti polinom helyettesítési értékei között zérus is lehet!

Ehhez programunkat gyökeresen át kell alakítani. Előbb az alábbiakat vegyük figyelembe:

– A mátrixot háromszor kell előállítani. Legyen az eredeti polinom helyettesítési értékeiből képzett mátrix $[a_{i,j}]_{49 \times 11}$, a csökkentett együtthatójú polinomból képzetté $[b_{i,j}]_{49 \times 11}$. Ekkor a harmadik mátrixot az előző kettőből az alábbiak szerint állítjuk elő:

$$\left[\frac{100b_{i,j}}{a_{i,j}} \right]_{49 \times 11}$$

A programban a három mátrixot rendre a **Z(I, J)**, **V(I, J)** és a **W(I, J)** tömbazonosítókkal fogjuk jelölni.

– A **Z(I, J)** mátrix tartalmaz zérus értékű elemeket, vagy esetleg olyan kicsi értékű elemeket, hogy a **W(I, J)** tömb hozzátartozó elemeit az adott mezőszélességen nem tudjuk elhelyezni. Ezért kiszűrjük a **Z(I, J)** túl kicsi elemeit és az ezeknek megfelelő százaléktételek előállítását megakadályozzuk. Ezt a feltétlen **GO TO** és aritmetikai **IF** utasítás együttes alkalmazásával érhetjük el. (OSZ: 128 és 131. oldal, és LV: 53. és 55. oldal.)¹

¹ Ajánlott irodalom : Obádovics–Szelecsán: Bevezetés a programozásba, valamint Lőcs–Vigassy : Fortran c. könyvek, melyeket a továbbiakban is OSZ, ill. LV-vel rövidítünk.

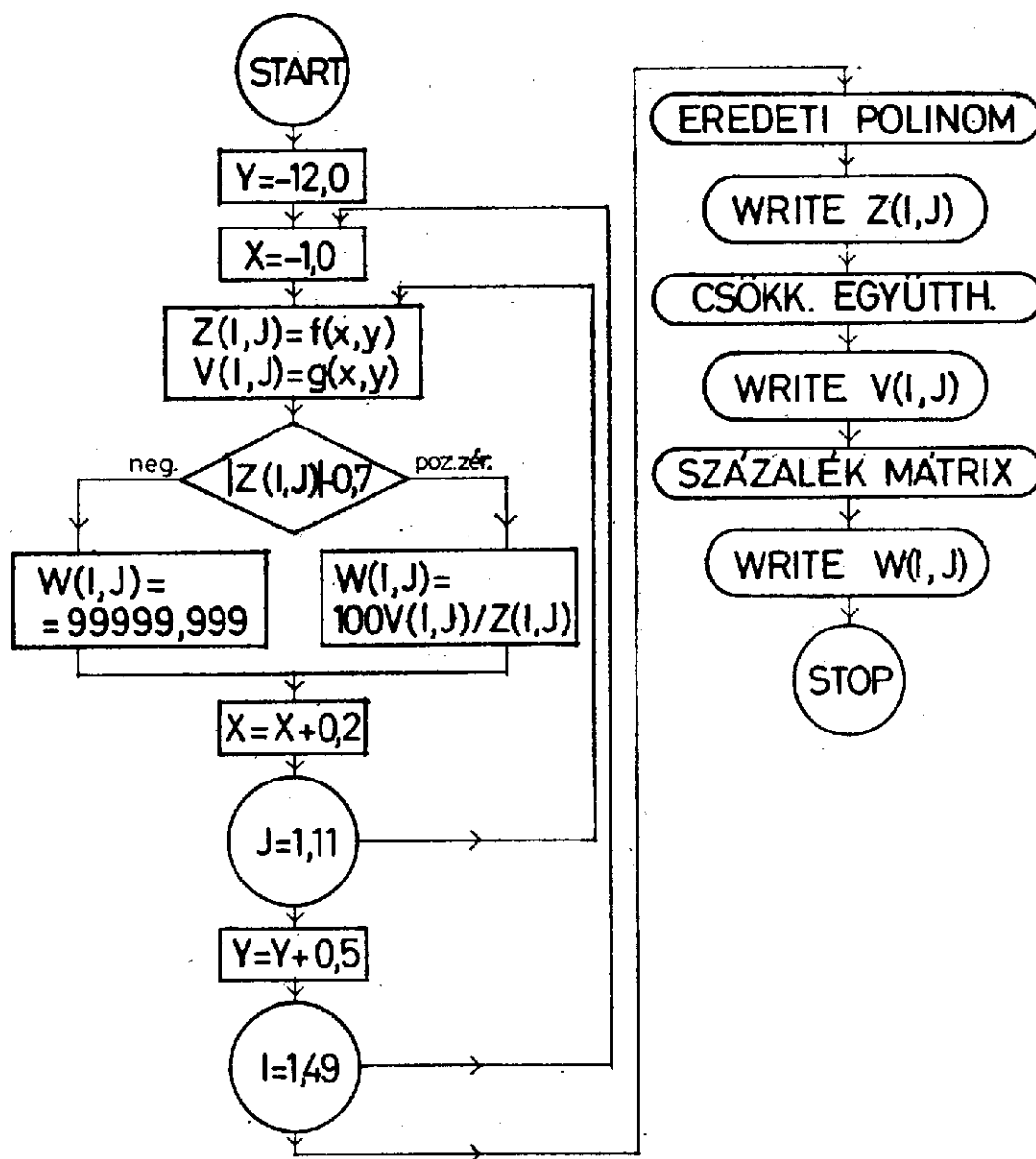
Kérdés, hogy a $Z(I, J)$ elemeinek abszolút értékei közül melyiktől kezdve kell megakadályozni a százalékszámítást. Vegyük alapul, hogy egy nyomtatási mezőn a legnagyobb elférő szám 99999, továbbá 999, hogy $V(I, J)$ elemeinek legnagyobb értéke sem haladja meg a 6000-et.

Könnyen kiszámítható, hogy a százalékszámítás alapjául megengedhető legkisebb érték szigorúan felfelé kerekítve 0,07. Ha tehát a $|Z(I, J)| - 0,07$ különbség negatívvá válik, akkor a százalékszámítást nem engedjük meg, helyette kinyomtatjuk a legnagyobb elférő számot. Ha ugyanis $Z(I, J)$ túl kicsi, akkor a számláló értéke nagyobbá válik a legnagyobb elférő számnál.

– Felmerülhet az a kérdés, hogy mi történjen az összetartozó, de ellentétes előjelű $Z(I, J)$ és $V(I, J)$ tömbelemekből képzendő százalékok esetén. Miután ilyenkor negatív számot kapunk, minden külön programozási fogás nélkül megállapodhatunk abban, hogy a kinyomtatott negatív $W(I, J)$ tömbelemeket az értékelésnél nem vesszük figyelembe.

– Az abszolútérték számítására az ún. **standard függvényazonosítók** egyike szolgál (OSZ: 120. oldal, LV: 34. oldal). Ilyen pl. a **SQRT** is, melyet korábban már használtunk. Egy X változó abszolút értékét az **ABS** függvényazonosító segítségével **ABS(x)** alakban állítjuk elő.

– A három mátrix megkülönböztethetősége érdekében mindegyik fölé szöveget nyomtatunk.



– A programhoz blokkdiagramot készítettünk. Ebben a feltételes ugrató utasítás blokkját egy rombusz jelöli (OSZ: 78. oldal), a két polinomot $f(x, y)$, ill. $g(x, y)$ jelöli (l. ábra).

Ezek után a program az alábbi módon írható:

```

MASTER PNEGY2
DIMENSION Z(49,11), V(49, 11), W(49, 11)
Y=-12.0
DO 1 I=1,49
X=-1.0
DO 2 J=1,11
Z(I,J)=X*(X*(2*X+6)-8*Y)+Y*(10-4*y)-12
V(I,J)=X*(X*(X+5)-7*Y)+Y*(9-3*Y)-11
IF (ABS(Z(I,J))-0.07)5, 0, 0
W(I,J)=100*V(I,J)/Z(I,J)
GO TO 2
5 W(I,J)=99999.999
2 X=X+0.2
1 Y=Y+0.5
WRITE (3,3)
3 FORMAT (1H1//10X,26HEREDETI EERTEEKTAABLAZAT://)
WRITE (3,9)(Z(I,J), J=1,11), I=1,49)
WRITE (3,6)
6 FORMAT (1H1//10X,33H1-EL CSOKKENTETT EGYUETTHATOOVAL)
WRITE (3,9)((V(I,J), J=1,11), I=1,49)
WRITE (3,7)
7 FORMAT (1H1//10X,18HSZAAZALEEK MATRIX:)
WRITE (3,9)((W(I,J), J=1,11), I=1,49)
9 FORMAT (//49(5X,11F11.3/))
STOP
END

```

Megfigyelhető, hogy ebben a programban három **WRITE** utasítás hivatkozik ugyanarra a **FORMAT**-ra.

4. Szubrutinok, szegmensek

A programozási gyakorlatban sokszor előfordul, hogy ugyanazt a programrészt más-más értékekkel újra kell számítani. Ilyen esetekre a programozási nyelvek ún. **szubrutinok** használatát teszik lehetővé. A **FORTTRAN** nyelv három szubrutin fajtájáról az utasításfüggvényről (OSZ-ben „függvényutasítás”), a **FUNCTION**-ról és a **SUBROUTINE**-ről OSZ könyve részletesen ír (165–179. oldal). A szubrutin olyan programrész, amelyet egy másik programrész „behív”, azaz aktivizál. A hívó program átad a szubrutinnak egy vagy több paramétert, a paramétereknek „pillanatnyi” úgynevezett aktuális értékeivel végzi el a szubrutin a benne foglalt programot. Ezeket meghatározott azonosítók tartalmazzák, minden behívásnál általában más paramétert. Ezután a szubrutin visszaadja az általa számított aktuális paramétereket az őt hívó programnak, amelyik azzal tovább dolgozik. Elsőnek a legegyszerűbb szubrutinnal foglalkozunk.

4.1. Az utasításfüggvény

Következő példánkban bemutatjuk az utasításfüggvény használatát. Számítandók a **PNEGY1** programban szereplő polinom helyettesítési értékei, ha $x = 1,9$ és $y = -0,8$; ha $x = -0,8$ és $y = 1,9$; ha x az első esetben, y a második esetben kapott függvényérték; ha x a második esetben, y az első esetben kapott függvényérték.

Egy lehetséges program az alábbiakban látható:

```

MASTER PNEGY3
POL(X,Y)=X*(X*(2*X+6)-8*X)+Y*(10-4*Y)-12
X=1.9
Y=-0.8
Z1=POL(X,Y)
Z2=POL(Y,X)
Z3=POL(POL(X,Y),POL(Y,X))
Z4=POL(Z2,Z1)
WRITE(3,2)
2 FORMAT (1H1////////)
WRITE (3,1)Z1,Z2,Z3,Z4
1 FORMAT (14X,3HZ1=,13X,3HZ2=,13X,3HZ3=,13X,3HZ4=//10X,4(F11.3,5X))
STOP
END

```

Az utasításfüggvényt, amit most **POL(X,Y)**-nal jelölünk, deklarálni kell. Ez azt jelenti, hogy megadjuk azt a függvényt, aminek a helyettesítési értéke lesz a **POL(X,Y)**, ezen a helyen azonban még nem számolja ki a program az értékét. Ha egyéb deklarációk is vannak, pl. **DIMENSION** stb., akkor azok után kell állnia, de az első végrehajtandó utasítás előtt.

Az utasításfüggvényt azonosítóval látjuk el, ez azonban nem egyezhet meg a standard függvényazonosítókkal (mint pl. **SQRT**, **ABS**, **SIN**, **COS** stb.) és más programbeli azonosítóval sem.

Az utasításfüggvény neve után zárójelben legalább egy paraméternek kell állnia. Ha a deklarációban szerepel a függvényutasítás, akkor ezeket **formális paramétereknek** nevezzük. Ha viszont végrehajtandó utasításban szerepel, akkor **aktuális paraméterekről** beszélünk. Az utasításfüggvényben a formális és aktuális paraméterek index nélküli azonosítók lehetnek. A paraméterek típusait a deklarációs szabályok határozzák meg, sorrendjük előírja a végrehajtható utasításokban aktuális paramétereinek sorrendjét.

A bemutatott programmal kapcsolatosan megjegyezzük, hogy a **Z3** azonosítóra vonatkozó értékadó utasítás csak az utasításfüggvény alkalmazási lehetőségeinek bemutatása érdekében ilyen összetett, egyébként célszerűbb lett volna a **Z4** azonosító értékadó utasításához hasonlóan készíteni. A programban az adatok fölé szövegrészek nyomtatását írtuk elő.

Feladatok

1. Készítsünk programot a négyzet alapú csonkagúlnak térfogatának számítására, ha az egyik négyzet területe minden számításnál egységnyi, a másik területe $1 \leq a^2 \leq 5$ között változik, $\Delta a^2 = 0,25$ lépésközzel. Minden lépésben nyolcféle magassággal kell a térfogatot számítani. Ezek:

$$\begin{aligned} m_1 &= a; & m_2 &= \sqrt{a}; & m_3 &= a^2 - 1; & m_4 &= \sqrt{a^2 + 1}; \\ m_5 &= \sqrt{1 + \sqrt{a}}; & m_6 &= \sqrt{|a^2 - a - 1|}; & m_7 &= \sqrt[3]{a^2}; & m_8 &= \frac{a + 1}{\sqrt{a + 1}}. \end{aligned}$$

Az oszlopok felett nyomtassuk ki az M betűt és közvetlen mellette az indexet. A térfogatokat négy tizedesjeggyel nyomtassuk. A program utasításfüggvény felhasználásával készüljön.

2. Készítsünk programot a kétismeretlenes egyenletrendszer megoldására másodrendű determinánsok alkalmazásával, melyeket utasításfüggvényben számíttatunk. Az

$$a_{11}x + a_{12}y = b_1, a_{21}x + a_{22}y = b_2$$

egyenletrendszert kell megoldani, ha

$$\begin{array}{lll} 1 \leq a_{11} \leq 10 & \text{és a lépésköz} & \Delta a_{11} = -1 \\ 8,8 \leq a_{12} \leq 10,6 & \text{és a lépésköz} & \Delta a_{12} = -0,2 \\ -6 \leq b_1 \leq 7,5 & \text{és a lépésköz} & \Delta b_1 = -1,5 \\ 1,7 \leq a_{21} \leq 4,4 & \text{és a lépésköz} & \Delta a_{21} = -0,3 \\ 2,6 \leq a_{22} \leq 9,8 & \text{és a lépésköz} & \Delta a_{22} = +0,8 \\ 0 \leq b_2 \leq 4,5 & \text{és a lépésköz} & \Delta b_2 = +0,5 \end{array}$$

A negatív lépésköz a felső korláttól történő csökkenést jelent. Az együtthatókat együtt, egyszerre változtatjuk a megadott lépésközök szerint, ezért 10 egyenletrendszert kell megoldani. A megoldáshoz a Cramer-szabályt alkalmazzuk. Ezzel és a továbbiakkal kapcsolatban javasoljuk SCHARNITZKY Viktor: MÁTRIXSZÁMÍTÁS című könyvének tanulmányozását. Mai feladatunkhoz a 7–13. oldalon találunk segítséget (Műszaki Könyvkiadó, Budapest, 1970; ára 25–Ft).

A program nyomtassa ki az egyenletrendszereket és tőlük jobbra két tizedesjegy pontossággal a megoldásokat, ha azok léteznek. Ha nincs pontosan egy megoldása, akkor a VÉGTELEN SOK MEGOLDÁS, ill. a NINCS MEGOLDÁS szöveg nyomtatandó, aszerint, hogy melyik eset áll fenn. A feladatokat a következő címre küldjék be megoldóink :

Gergely János
MŰM Számítástechnika Intézet

1089 Budapest
Reguly Antal u. 57–59.