

Előző számunkban megkezdjük a **FORTRAN** nyelv néhány kiválasztott specifikációjának ismertetését. Először az ezzel kapcsolatban kitűzött feladat megoldásairól néhány szót.

Mint említettük, a papír teljes szélessége 128 karakterpozíció, amiből adódik, hogy 45 az a legnagyobb szám, ami a sorban még nyomtatható. Bemutatunk egy lehetséges megoldást.

Ebben a programban **J** egy 45 elemű, egydimenziós tömb. (Indexes változókról OSZ : 109. old., LV : 101. old.) A ciklus ezt a tömböt betölti. A nyomtató utasítás az egész tömbre vonatkozik.

```

1 | MASTER TER1
  | INTEGER J(45)
  | DO 1 I=1,45
  | J(I)=I
  | WRITE (3,2) J
2 | FORMAT (912,3613)
  | STOP
  | END

```

Megjegyzés: Ha a **J** egyetlen egy szám azonosítója lett volna, akkor az automatikus típusdeklaráció (OSZ: 112. old., LV : 100, old.) értelmében deklarálása fölösleges lett volna. A tömb deklarálása kötelező! Ez másképpen is történhet, pl. így:

```

| | DIMENSION J (45)

```

és ekkor a típusdeklaráció szabályával összhangban ez automatikusan egész típusú tömb.

További kiegészítő megjegyzés: a **FORMAT** utasításnak nem kell követnie a rá hivatkozó **WRITE** utasítást, az adott programban bárhol elhelyezkedhet a **MASTER** és a hozzá tartozó **END** között.

Folytatjuk a specifikációk ismertetését

3.2. Az **F** specifikáció valós típusú számok kiíratására szolgál. A szám nyomtatási képe mindig tartalmaz tizedes-pontot. A specifikáció alakja:

F*w.d*

ahol *w* és *d* természetes számok. *w* adja meg a szám részére igényelt mezőszélességet (beleértve az esetleges negatív előjelet és a tizedespontot), *d* adja meg, hogy a kiíratandó szám hány tizedesjegyre legyen kerekítve. Például a

```

7 | Y=2.
  | GYOEK=SQRT (Y)
  | WRITE (3,7) Y,GYOEK
  | FORMAT (2F7.3)

```

programrészlet hatására (**SQRT**-ről lásd OSZ : 120. old., LV: 35. old.) a nyomtatási kép:

▽▽2.000▽▽1.414

Ebből láthatjuk, hogy az egész rész nem értékes számjegyei helyén space áll. akkor mi legyen? Ha pl. egy változó értéke 0.8366600265 és **F10.8** specifikációval kerül nyomtatásra, akkor 0.83666003 jelenik meg a papíron.

Mintafeladat: készítsünk programot, amely az $\frac{x^3 - 2x}{x^2 + x - 0,1}$ kifejezés helyettesítési értékeit öt tizedesjegyre kerekítve számítja és nyomtatja a benne szereplő változó -1.783 , $0,861$ és $1,097$ értékei mellett.

Megoldás: deklaráljunk két háromelemű, valós típusú tömböt. Az egyik tömb elemei felveszik az adott számértékeket, majd egy ciklusban kiszámítjuk a helyettesítési értékeket. Ezek lesznek a másik háromelemű tömb komponensei.

```

1 | MASTER P1
  | DIMENSION X(3), Y(3)
  | X(1)=-1.783
  | X(2)=0,861
  | X(3)=1,097
  | DO 1 I =1,3
  | Z=X(I)
1 | Y(I)=(Z*(Z*Z-2.))/(Z*(Z+1)-0.1)
  | WRITE (3,2) X,Y
2 | FORMAT (3F10.3,3F12.5)
  | STOP
  | END

```

A futtatás eredményeként az alábbi output nyerhető:

▽▽▽▽-1.783▽▽▽▽▽0.861▽▽▽▽▽1.097▽▽▽▽ - 1.62205▽▽▽▽-0.72137▽▽▽▽-0.39714

3.3. A **P1** program nyomtatási képe az áttekinthetőség szempontjából nem a legjobb. Bemutatjuk azokat a specifikációkat, amelyekkel a soron belül kihagyásokat, soremelést és lapváltást hozhatunk létre.

3.3.1. **Helykitöltés.** Üres nyomtatási helyet (helykitöltést) az **I** és **F** specifikációk túlméretezett mezőszélességeivel már elértünk. Célszerűbb azonban, ha üres helyet a

$w\mathbf{X}$

alakú specifikációval állítunk elő. Itt a w természetes szám, amely megadja, hogy hány karakterpozíciót hagyjon ki a sornyomtató. A P1 példában a

2| |FORMAT (3F10.3,10X,3F12.5).

utasítást alkalmazva (1. variáns) olyan nyomtatási képet kapunk, amelyen a harmadik és negyedik nyomtatási mező között 10 üres hely áll. A tényleges space-ek száma ekkor a harmadik és negyedik szám között 15 lesz.

3.3.2. **Soremelés.** A **FORMAT**-on belül a specifikációk előtt, után, ill. között elhelyezett / (azaz per-) jel soremelést eredményez, az ezt követő specifikáció utáni rész a nyomtatásban a következő sor elejére kerül. Ha a P1 példában a **FORMAT** az alábbi lenne (2. variáns)

| |FORMAT (5X,3F12.3/5X,3F12.5),

akkor a nyomtatási kép az alábbi lenne:

▽▽▽▽▽▽▽▽▽▽-1.783▽▽▽▽▽▽▽▽0.861▽▽▽▽▽▽▽▽1.097
▽▽▽▽▽▽▽▽▽▽-1.62205▽▽▽▽-0.72137▽▽▽▽ - 0.39714

A per jel alkalmazásával és azzal, hogy mindkét sorban azonos mezőszélességet írtunk elő, elértük, hogy a két sorban az összetartozó értékek egymás alá kerültek.

Megjegyzendő, hogy

- a per jel vesszőt helyettesíthet, azaz két specifikációt elválaszthat egymástól;
- több per jel is írható egymás mellé, valamennyi külön-külön soremelést idéz elő;
- viszont a **FORMAT**-on belül két vessző nem követheti közvetlenül egymást.

Formailag még jobb nyomtatási képet kaphatunk, ha a **P1** programba 3. variánsként a következő **FORMAT** utasítást írjuk:

2| |FORMAT (/////46X,3F12.5//46X,3F12.5).

Itt figyelembe vettük, hogy a három mező együttes szélessége 36 karakterpozíció, ezt levontuk a teljes lapszélességből, a maradékot megfeleztük és így kaptuk a 46 karaktert, amit a sor elején üresen hagyunk. A nyomtatás most a lap közepére kerül, a számok tizedespontjai egymás alatt lesznek, és az első sort sem a lap legfelső sorába nyomtatjuk, hanem hat sorral lejjebb, az esztétikus hatás kedvéért.

3.3.3. Eddig ugyan nem említettük, de minden eddig tárgyalt nyomtatási utasításra igaz, hogy bármely sor első karakterpozícióján a nyomtatás lehetetlen. Ugyanis az első karakterpozícióra írt karakter a sornyomtató gépet magát vezérli. Úgy is mondhatjuk, hogy az első karakter a sorban egy kód, aminek hatására a gép valamilyen speciális tevékenységet végez. Emiatt a sor első karakterét **vezérlő karakter**-nek is szoktuk nevezni. Mi az alábbi vezérlő karaktereket fogjuk használni:

- zérus szám, melynek hatása dupla soremelés;
- egyes szám, melynek hatása egész lapváltás (kezdés a következő lap első sorában).

Kérjük, hogy a részünkre beküldendő programjaitokban más, pl. az LV könyvben leírt vezérlő karaktereket ne használjátok, mert egyesek közülük – programozási hibával párosulva – elronthatják a sornyomtatót.

Nyilván felmerül a kérdés, hogy az első karakterpozícióra hogyan írunk vezérlő karaktert. Erre egy lehetőséget mutatunk, előrebozsátva, hogy nem ez az elterjedt, legcélszerűbb módszer. (Arra még visszatérünk.)

5| |K=1
| |WRITE (3,5)K
| |FORMAT (11)

E programrészlet végrehajtásakor az első pozícióra kerülő 1-es számjegyet a gép vezérlő karakternek értelmezi, melynek hatására új lapot kezd (az 1-es számjegy nem kerül kinyomtatásra).

Megjegyezzük, hogy ha ebben a programrészben **K**-nak 154 értéket adtunk volna és a **FORMAT**-ban **13** specifikáció állna, akkor az eredmény lapváltás lenne, és az új lapon az 54 szám lenne kinyomtatva a második és harmadik karakterpozíciókon.

3.3.4. A **FORMAT** utasítás első és utolsó zárójele között áll a **format lista**. A format lista specifikációk sorozata. Sok esetben hasznos, ha az ismétlési tényező nem egyetlen specifikációra vonatkozik, hanem azok egy csoportjára. A

format listán belül erre úgy adódik lehetőség, hogy az ismételni kívánt specifikáció-csoportot zárójelbe tesszük és a zárójel elé írjuk az egész csoportra vonatkozó ismétlési tényezőt. Ha pl. a

13| |FORMAT (10X,3(3X,213,4X,3F5.2))

utasítást használjuk, akkor tíz space után háromszor egymás után nyomtatva lesz két egész és három valós szám. Az erre hivatkozó **WRITE** utasításban olyan azonosítóknak kell lenni, melyek összesen 15 elemet tartalmaznak, mivel a példánk is tizenöt szám specifikációját tartalmazza. (Természetesen az azonosítók típusainak rendre meg kell egyezniük a specifikációk típusaival.) A megismételt listarész soresmelést, lapváltást is tartalmazhat. Az ismétlések többször, egymásba skatulyázva is szerepelhetnek a format listában.

P1 jelű programunkban az utolsó variáns **FORMAT** utasítása rövidebben így írható, az elmondottak alapján (4. variáns):

2| |FORMAT (///2/(46X,3F12.5/)).

3.4. Sok esetben nemcsak eredményeknek, hanem betűkből, írásjelekből, számjegyekből álló szövegrészeknek a kinyomtatására is szükségünk lehet. Például táblázati fejlécek, mértékegységek stb. feltüntetésének esetében. Szövegek kiírására szolgál a **H** specifikáció (Hollerith név kezdőbetűje.) A specifikáció alakja:

$w\mathbf{H}$

ahol w a mezőszélesség meghatározására szolgáló természetes szám. A **H** után álló w számú karakter, az ún. Hollerith mező a kinyomtatandó szöveget tartalmazza. Például a

23| |FORMAT (5X,17H ∇ GULA ∇ TERFOGATA=,F5.2,8H ∇ ∇ KOEBDM)

utasítás hatására (ha az eredményül kapott szám pl. 62.15)

$\nabla\nabla\nabla\nabla\nabla$ A ∇ GULA ∇ TERFOGATA = 62.15 $\nabla\nabla$ KOEBDM

sort nyomtatja a gép. Nagyon ügyeljünk arra, hogy a **H** után a szöveget alkotó karakterek száma pontosan megegyezzen a **H** előtt álló mezőszélességgel. Megjegyzendő még, hogy a **H** specifikációhoz nem tartozik **WRITE** után álló listaelem. Fenti példánk esetében pl.

WRITE (3,23) VOL

nyomtató utasítás tartozhat a **FORMAT**-hoz, amelyben két **H** specifikáció és egy **F** specifikáció van.

A **H** specifikáció kényelmes lehetőséget nyújt lapváltás előírására. Ha a sor elejére **1H1** listaelemet írunk, akkor az lapváltást eredményez. Pl. a

FORMAT (1 H1, ... /1 H1, ...)

utasításban két lapváltást írtunk elő. Látható, hogy az **1H1** a **FORMAT** belső részében csakis akkor idéz elő lapváltást, ha közvetlenül perjel után áll.

Feladat

Készítsünk programot, amely kinyomtat egy sakktáblát úgy, hogy azon a fekete négyzeteket **X** betűvel töltjük ki. Egy négyzet egy sorban hat karakterpozícióból álljon, és egymás alatt négy ilyen sor legyen. A 6-szor 4 karakter alkot egy négyzetet. A sakktábla a papírlap közepére kerüljön.