

Megkezdjük számítástechnikai példáink, feladataink közzétételét. Mint azt már az előző számunkban jeleztük, a FORTRAN programozási nyelvet fogjuk használni. Olvasóinknak a programozási nyelv megismeréséhez a következő két könyvet ajánljuk:

Obádovics- Szelezsán: Bevezetés a programozásba (Tankönyvkiadó, ára 12 Ft).

Lőcs – Vigassy: FORTRAN programozási nyelv (Műszaki Könyvkiadó, ára 26 Ft).

Az első könyv, (amelyet a továbbiakban OSZ rövidítéssel jelölünk), kezdőknek való, nem tartalmaz minden szükséges ismeretet. A második könyv, (melyet a továbbiakban LV rövidítéssel jelölünk) a részünkre szükségesnél sokkal többet tartalmaz, nehezen áttekinthető, ezért azok részére, akik ezt a könyvet kívánják használni, közvetlen kapcsolatfelvétel esetén megadjuk, mely részeket, milyen sorrendben olvassanak el.

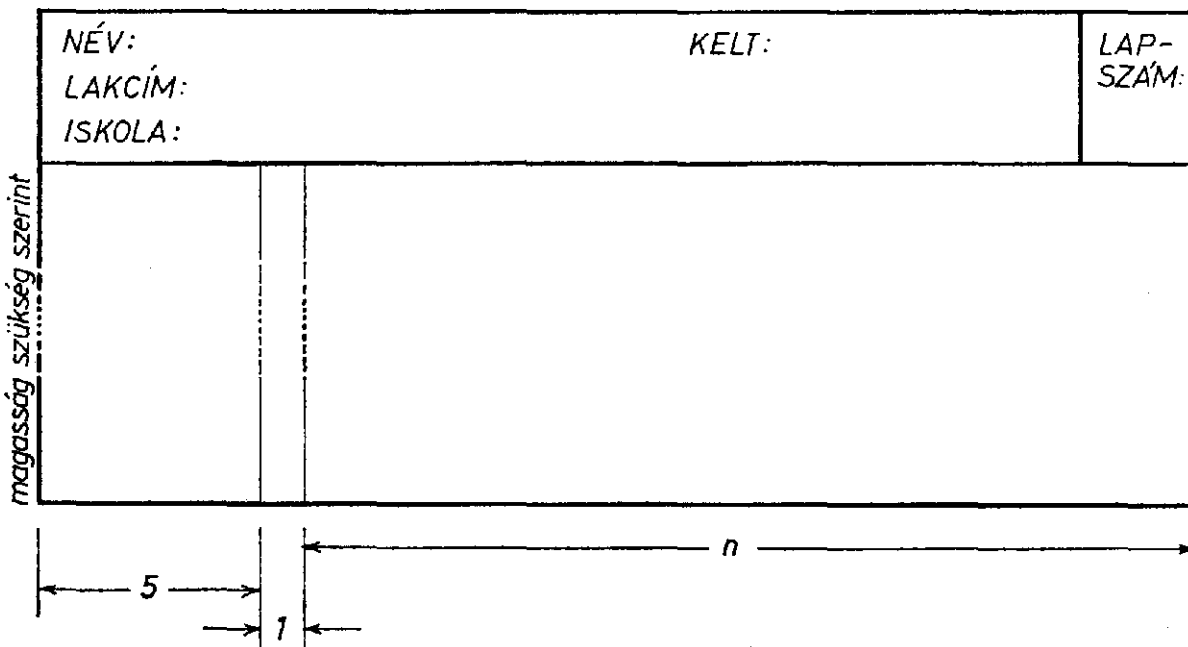
A Számítástechnikai Rotatot a Munkaügyi Minisztérium Számítástechnikai Intézet KISZ kollektívája készíti dr. Ada-Winter Péter irányítása mellett. A számítástechnikai tárgyú feladatmegoldásokat, kérdéseket, kéréseket az alábbi címre kérjük küldeni :

MŰM Számítástechnikai Intézet
Gergely János
1089 Budapest
Reguly Antal u. 578212;59.

Minden megküldött programra választ adunk, hibás programok hibáit megjelölve visszaküldjük, szükség esetén magyarázattal. Jó programok megküldése esetén a feladatmegoldók nevét közöljük, vagy írásban értesítjük őket. Egyes kiváló feladatmegoldások esetén a beküldött programot számítógépen futtatjuk és a kinyomtatott lapot (outputot) a készítőjének megküldjük.

1. A programkészítés formai szabályai

Programjainkat, utasításainkat programlapra kell írunk. A programlap-nyomtatványok pénzért is beszerezhetők, de egyszerűbb és olcsóbb, ha magatok készíttétek el. Vegyetek egy „franciakockás8221; papírt (szükség esetén a négyzethálós is megteszi), húzzatok rá keretvonalat és a kereten belül osszátok be az 1. ábrán látható módon. (Négyzethálós papír esetében a hosszabb oldal mentén húzzátok meg a fejlécezt.)



1. ábra

A program minden utasítása a lapon levő tégl- vagy négyzetháló külön sorába írandó. Egy-egy sorban a háló által meghatározott területeket **karakterpozícióknak** nevezzük. Az első öt karakterpozíció alkotja a **cimkemezőt**. A hatodik karakterpozíciót nem fogjuk használni. A hetedikről kezdődő pozíciók alkotják az ún. **utasításmezőt**, amely legfeljebb 68 pozícióból állhat. (A nyomtatott programlapokon ugyan 74 pozíció áll, de az utolsó 6-ra már nem írható utasítás.) A programlapokon 8212; a fejléctől eltekintve 8212; csakis a FORTRAN nyelvben megengedett jelek szerepelhetnek (OSZ: 105. old. , LV: 15).

Kérjük az alábbi szabályok figyelembevételét :

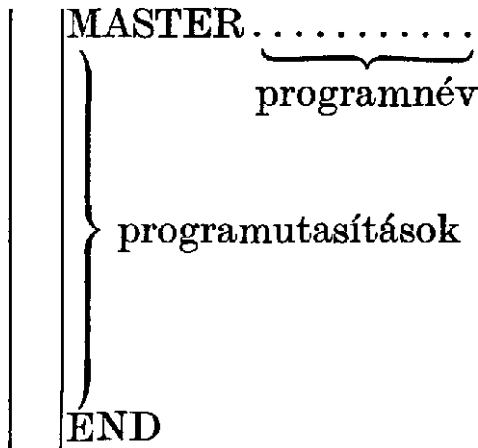
- Minden betű, szám, írásjel (közös névvel: karakter) külön karakterpozícióba kerüljön.
- A programlapokon a betűket csakis nagybetűkkel szabad írni.
- Az 0-betűinek a zérus számtól való megkülönböztetése érdekében a programlapokon a zérus szám egy ferde vonallal áthúzandó, így : 0.

Nem szabad a papírlap mindkét oldalát használni. Ha a program egy lapon nem fér el, új lapot kell nyitni és számozni kell a lapokat.

Végül kérjük, hogy a programba **comment** sorokat ne írjatok (OSZ: 198. old. , LV: 14. old.), mert ezzel sok többletmunkát okoztok a társadalmi munkában dolgozó kártyalyukasztó lányoknak.

2. Programjaink szerkezetéről

Először csak egészen egyszerű programokat fogunk készíteni. Az ilyen programok szerkezete az alábbi (a két függőleges vonal a 6, üresen hagyandó karakterpozíciót zárja közre):



Minden program az ún. MASTER utasítással kezdődik, amiből egy programban csak egy lehet. A MASTER szó után egy szóközt hagyva írhatjuk az általunk kitalált programnevet. A programnévnek betűvel kell kezdődnie, számjegyen és betűn kívül más nem tartalmazhat, és legfeljebb hat karakterből állhat. A program utolsó utasítása az END utasítás. E két utasítás között helyezkednek el a program többi utasításai. Megjegyezzük, hogy a program futását a számítógépen a STOP utasítás állítja meg (OSZ: 106. old., LV: 90. old.), melyből a programban legalább egynek kell szerepelnie, ég amely a programon belül bárhol elhelyezkedhet.

3. A kiíratás utasításai

A program végrehajtása során az ún. változók, más néven azonosítók, számértékeket vchctnck fcl (OSZ: 109. old. , LV: 16. old.). Ezek egy részét mi eredménynek tekintjük, ezért ki akarjuk írni a számítógép sornyomtatóján (OSZ: 57. old.). A FORTRAN nyelv nyomtató utasítására vonatkozó legfontosabb ismereteket itt kell elmondanunk, mert azok az OSZ könyvben túlságosan leegyszerűsítettek, az LV könyvben viszont a mi számunkra túlrészletezett. A nyomtatás tulajdonképpen két utasításból áll. Ezek a WRITE és a FORMAT, melyek közül az első azt adja meg, hogy a végrehajtás során mikor és mit kell kinyomtatni, míg a második azt írja elő, hogy hogyan, milyen formában történjen a nyomtatás. A WRITE utasítás alakja a programlapon:

```
| | WRITE (u, f) S1, S2, MAG, HOSSZ
```

ahol u és f természetes számok. Jelentésük :

– u neve **logikai perifériaszám**, feladata meghatározni, hogy melyik perifériális egységen (OSZ: 41. és 15. old.) történjen a kiíratás. A mi számítóközpontunkban az általunk használandó sornyomtató perifériaszáma 3, ezért feladatokban u helyére mindig 3-at kell írni.

– f egy ún. **címke** (OSZ: 127–128. old. , LV: 14. old.) , amely azt mutatja meg, hogy melyik az a FORMAT utasítás, amelyik ehhez a WRITE utasításhoz tartozik. (Minden WRITE-hoz pontosan egy FORMAT tartozik, de fordítva nem ; egy FORMAT tartozhat több WRITE utasításhoz is.)

– bemutatott példánkban a zárójel után négy olyan azonosító áll, amelyek értékét kiíratjuk. Az azonosítóknak itt elhelyezkedő sorát közös névvel **output listának** nevezzük.

A FORMAT utasítás előtt mindig áll egy címke, amely azt egy vagy több WRITE utasításhoz rendeli. A FORMAT alapszó után két gömbölyű zárójel között állnak az ún. **specifikációk**. Egy specifikációval egy számadatot vagy szöveget jelentő karaktercsoport helyezhető el a nyomtatási sor egy intervallumába, az u . n. **mezőbe**. A mező hosszát **mezőszélességnek** mondjuk, jelölésére a w betűt használjuk. (w a mezőn elhelyezkedő karakterpozíciók száma.)

A következőkben részletesebben megismertetjük az általunk használni kívánt specifikációkat.

3.1. Az I specifikáció egész típusú (integer) számok kiíratására szolgál. A FORMAT-on belül Iw alakban szerepel, ahol w jelenti azon karakterpozíciók számát, amelyeken egy kiírandó egész számot nyomtatni kívánunk. Például a következő programrészlet

```
| | IND =128
| | WRITE(3,8)END
8 | | FORMAT(I5)
```

hatására a mezőben kiírt karakterek és betűköz (∇ a betűköz, más néven space jele, ami természetesen nem jelenik meg a nyomtatón:

$\nabla \nabla 128$

Ha IND értékül 1289-et adtunk volna, akkor a mezőben

$\nabla 1289$

lett volna. Szabály, hogy w értéke a nyomtatandó szám számjegyeinek számánál nem lehet kisebb. Ha negatív számot írunk ki, akkor annál legalább eggyel nagyobbak kell lennie, az előjel helyének biztosítása érdekében. Tömbazonosító használatánál (OSZ: 193. old. , LV: 101. old.) célszerű a specifikáció elé **ismétlési tényezőt** írni. Ezzel például az $I5, I5, I5$ specifikációsorozatot $3I5$ formájúvni rövidíthetjük. Ha n az ismétlési tényező, akkor a

| | FORMAT(nIw)

specifikációval egy sorba n darab, egyenként w mezőszélességen elhelyezkedő egészet nyomtathatunk. Ügyeljünk arra, hogy a sornyomató szélessége 128 karakter, így hosszabb sorokat nem tudunk nyomtatni.

Példaként egy teljes programot mutatunk be, amely 3 első hat többszörögét nyomtatja . (A programban használt DO utasításról OSZ a 146., az LV 69. oldalon tájékoztat.)

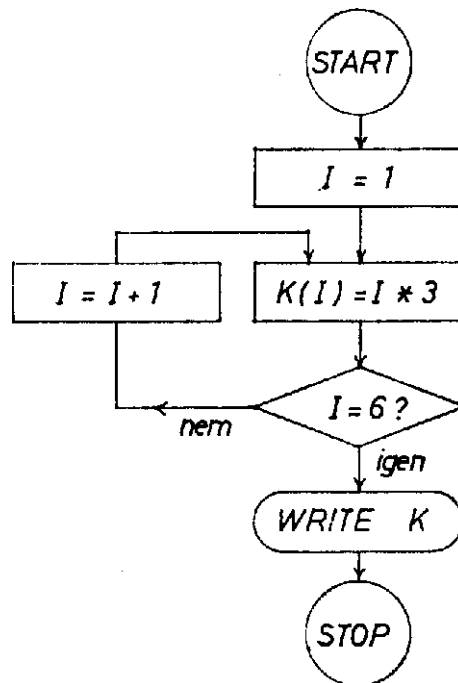
	MASTER KUTYA
	DIMENSION K(6)
	DO 10I=1,6
10	K(I) = I * 3
2	WRITE(3,20)K
20	FORMAT(6I5)
	STOP
	END

A program outputján a mező elhelyezkedése, tartalma a következő :

$\nabla \nabla \nabla \nabla 3 \nabla \nabla \nabla \nabla 6 \nabla \nabla \nabla \nabla 9 \nabla \nabla \nabla \nabla 12 \nabla \nabla \nabla \nabla 15 \nabla \nabla \nabla \nabla 18$

A futást könnyen el tudjuk képzelni, ha egy ún. **folyamatábrát** (blokkdiagramot) készítünk róla. Ezt láthatjuk a 2. ábrán. A folyamatábra startblokkja után a DO utasítás az I ciklusváltozónak az 1 számértéket adja. A következő blokkban a K tömb I-edik komponense kapja I háromszorosának értékét. Egy újabb blokkban a DO utasítás megvizsgálja az I értékét, és ha $I < 6$, 1-gyel növeli, és újra kezdi a ciklust; ha $I = 6$, a DO utasítás befejeződik és újabb utasítással kiíratjuk a tömböt, majd a program futását megállítjuk.

A specifikációk leírását következő számunkban folytatjuk.



2. ábra

Feladat: Készítsünk olyan programot, amely a lap egyetlen , 128 pozíciószélességű sorában 1-től növekvő rendben kinyomatja a természetes számokat úgy, hogy két szám között mindig csak egy space legyen:

$\nabla 1 \nabla 2 \nabla 3 \nabla 4 \nabla 5 \nabla 6 \nabla 7 \nabla 8 \nabla 9 \nabla 10 \nabla 11 \nabla 12 \nabla 13 \nabla \dots$