

# Házi feladat

Pere László

2004. január 17.

## Kivonat

Ez a dokumentum bemutatja hogyan készíthetjük el olvashatóan a házi feladatokat. A dokumentum  $\text{\LaTeX}$  segítségével készült, a forrása letölthető a <http://linux.pte.hu/progmodsz/címről>.

A példa közvetlenül felhasználható a *Programozás módszertan* tantárgy házi feladatainak elkészítéséhez.

## Tartalomjegyzék

1. Prímteszt	1
2. Faktoriális	1
3. Elöltesztelő ciklus átalakítása	1
3.1. Megvalósítás	1
4. Függvény ábrázolása	1

## 1. Prímteszt

A feladat olyan FORTRAN77 program elkészítése, amely egy változóban tárolt számról eldönti, hogy az prímszám-e.

```
1 PROGRAM PRIMSZAM
2 LOGICAL PRIM
3 N = 103
4 I = 2
5 PRIM = .TRUE.
6
7 50 IF (N.EQ.I*(N/I)) PRIM = .FALSE.
8 IF ((I.GT.N/2) .OR. (PRIM.EQV..FALSE.)) GOTO 100
9 I = I + 1
10 GOTO 50
11 100 CONTINUE
12 IF (PRIM.EQV..TRUE.) PRINT*, N
13 END
```

A program a 7–11. sorban egy elöltesztelő ciklust tartalmaz, amely megvizsgálja, hogy az  $N$  osztható-e  $I$ -vel ( $2 \geq I \geq N/2$ ).

A program a 12. sorban megvizsgálja, hogy azért fejeződött-e be a ciklus, mert a szám osztható volt a ciklusváltozó valamelyik értékével, vagy azért, mert elértük a ciklus felső határát. Ha elértük a ciklus felső határát, a szám prímszám, azt a program kiírja.

## 2. Faktoriális

Az  $n!$  kiszámításának rekurzív definíciója:

$$n! = \begin{cases} 1 & \text{if } n = 1 \\ n(n-1)! & \text{otherwise} \end{cases} \quad (1)$$

## 3. Elöltesztelő ciklus átalakítása

Alakítsuk át a következő algoritmusrészletet hátultesztelő ciklus használatára!

```
r = p%q
p = q
q = r
while q ≠ 0 do
  r = p%q
  p = q
  q = r
wend
```

Az algoritmus hátultesztelő ciklussal leírva a következő lesz:

```
repeat
  r = p%q
  p = q
  q = r
until q = 0
```

### 3.1. Megvalósítás

Az algoritmus megvalósítható a While nyelvén a következő program segítségével:

```
1 p := 876; q := 8687
2
3 repeat
4   r := p % q
5   p := q
6   q := r
7 until q == 0
8
9 print p
```

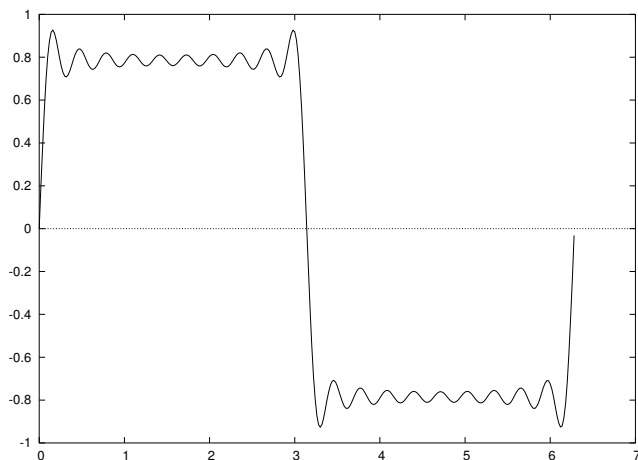
## 4. Függvény ábrázolása

Készítsük el annak a periódikus jelnek a képét, amelyben csak a páratlan felharmonikusok találhatók meg az  $\frac{1}{x}$  függvénynek megfelelően csökkenő amplitúdóval! A számítást végezzük 10 összetevőig.

A függvény értékét kiszámíthatjuk a következő formula alapján:

$$f(\omega) = \sum_{i=0}^N \frac{\sin(2i+1)\omega}{2i+1} \quad (2)$$

A feladat megoldása során először elkészítünk egy FORTRAN programot, amely kiírja a szabványos kimenetre  $\omega$  és  $f(\omega)$  értékeit a  $0 \dots 2\pi$  tartományban. Ezt a programot úgy futtatjuk, hogy a szabványos kimenetét átírányítjuk a `fuiggveny.text` állományba, amely két oszlopos ASCII állomány lesz.



1. ábra. A negyszogjel.f eredménye

Ez után elkészítünk egy egyszerű GNUplot programot, amely az állományt ábrázolja (ez a program mintául szolgálhat a házi feladatok megoldásához, hiszen a GNUplot nem a tananyag része).

A függvény értékeit kiszámító FORTRAN program a következő:

```

1      PROGRAM MAIN
2      C Az egyszerűség kedvéért legyen most I valós.
3      REAL I
4      DO 20 X=0, 6.28, 0.02
5          Y = 0
6          DO 10 I=0, 9
7              Y = Y + 1/(2*I+1)*SIN( X*(2*I+1) )
8          10 CONTINUE
9          PRINT*, X, Y
10         20 CONTINUE
11         END

```

A program a 4–10. sorban tartalmazza azt a DO ciklust, amely az  $\omega$  (itt X) tengely értékeit bejárja. Ezen a cikluson belül a 6–8. sorban lévő belső ciklus adja össze az egyes felharmónikusok értékeit.

A függvény ábrázolását végző GNUplot program a következő:

```

1      set terminal postscript
2      set output "fuggveny.ps"
3
4      set zeroaxis
5      set nokey
6      plot "fuggveny.text" using 1:2 notitle with lines

```

A függvény a 1. ábrán látható. Az ábrát a  $\text{\LaTeX}$  állományba az `\includegraphics` parancs segítségével töltöttük be. Ez kissé nehézkes, mivel fogatni és méretezni kell a képet, hogy a kéthasábos szedésben jól jelenjen meg. A  $\text{\LaTeX}$  kódrészlet a következő volt:

```

1      \begin{figure}
2      \centering{\rotatebox{-90}{\scalebox{.35}
3      {\includegraphics{fuggveny.ps}}}}
4      }
5      \caption{A {\tt negyszogjel.f}
6      eredménye\label{fig:negyszogjel}}
7      \end{figure}

```

Más módszereket – más eszközöket – használva a kép készítése és beillesztése esetleg egyszerűbb lehet. A keresőprogramok segítségével az Interneten meglehetősen sok példa található a GNUplot használatára, amelyek alapján összetettebb grafikonok is előállíthatók.

## Irodalomjegyzék

1. NYÉKYNÉ GAIZLER JUDIT (szerk.): *Programozási nyelvek*, Kiskapu Kiadó, 2003.
2. PERE LÁSZLÓ: *Bevezetés a  $\text{\LaTeX}$  használatába*, 2002.
3. PERE LÁSZLÓ: *Linux: felhasználói ismeretek I*, Kiskapu Kiadó, 2002.
4. TIMOTHY VAN ZANDT: *Documentation for 'fancyvrb': Fancy Verbatims in  $\text{\LaTeX}$* , CTAN:macros/latex/contrib/supported/fancyvrb.
5. Az algorithms  $\text{\LaTeX}$  csomag dokumentációja.