INFORMATIKA LABORATÓRIUM I.

Alapok, operátorok, feltételes utasítások

Operátorok

- Kétfajta operátort különböztetünk meg:
  - egy operandusú
  - két operandusú
- Egy operandus esetén az operátor lehet prefix és postfix:

```
operátor operandus
operandus operátor
```

- Két operandusú operátor esetén:

```
operandus1 operátor operandus2
```

Aritmetikai műveletek

```
+   összeadás
-   kivonás
*   szorzás
/   osztás
%   maradék képzés
```

Relációs műveletek

```
>
<
>=
<=
==   Egyenlő
!=   Nem egyenlő
```

Az eredményük vagy 0 (hamis) vagy 1 (igaz)

Logikai műveletek

&&	ÉS
||	VAGY
!	NEM

Kiértékelés balról jobbra. Ha a kifejezés értéke meghatározható, akkor a kiértékelés leáll.

Értékadás „okosan”

```
a = a + 1   ->   a += 1
a = a - 1   ->   a -= 1
a = a * 1   ->   a *= 1
a = a / 1   ->   a /= 1
```

Növelő és csökkentő operátorok

```
++
--
```

```
a++ -> a = a + 1;
++a -> a = a + 1;
```

Ha postfixként használjuk:

```
a = 15;
b = a++; // -> b = a; a = a + 1;
```

Ha prefixként használjuk:

```
a = 15;
b = ++a; // -> a = a + 1; a = b;
```

Bitenkénti logikai operátorok

&	bitenkénti ÉS
|	bitenkénti VAGY
^	bitenkénti kizáró vagy
<<	bitléptetés balra
>>	bitléptetés jobbra
~	egyeskomplemens

Precedencia

()	zárójel
! ++ – -	negálás, növelés, csökkentés, előjel
* / %	szorzás, osztás, maradékképzés
+ -	összeadás, kivonás
« »	bitenkénti eltolás balra és jobbra
< <= >= >	kisebb mint, kisebb vagy egyenlő, nagyobb vagy egyenlő, nagyobb mint
== =!	egyenlő, nem egyenlő
&	bitenkénti megengedő (inkluzív) és
^	kizáró (exkluzív) vagy
|	bitenkénti vagy
&&	és
||	vagy

Ha a egy művelet azonos szinten van, akkor az operátorok balról jobbra lesznek kiértékelve.
A zárójel mindig módosítja a kiértékelés sorrendjét.

Formátumozott kivitel

- A printf() függvény formátumozott kivitelt tesz lehetővé számunkra.
- A formátum kétféle karaktert tartalmazhat: amely kiíródik változás nélkül a kimenetre, és amely a soron következő argumentum konverzióját írja elő.
- Minden konverziós szakasz a % jellel kezdődik. Próbáljuk ki a következőket:

```c
#include <stdio.h>
main()
{
        double a = 31.123456;
        printf("%.2f\n", a);

}
```

```c
#include <stdio.h>
main()
{
        double a = 31.123456;
        printf("%20.2f\n", a);

}
```

```c
#include <stdio.h>
main()
{
        double a = 31.123456;
        printf("|%-20.2f|\n", a);

}
```

Formátumozott kivitel

A % jel és a konverziós karakter között a sorrendben a következők l ehetnek:

- mínusz jel, ami konvertált argumentumot balra igazítja
- egy szám, ami megadja a minimális mezőszélességet
- egy pont, ami elválasztja a mezőszélességet a pontosságot meg adó számtól
- egy szám (pontosság)
  - karaktersorozatnál a kiírandó karakterek maximális számát
  - lebegőpontos számok esetén a tizedespont után kiírt számjegy ek számát
  - egész karakterek esetén a kiírt számjegyek maximális számát
- egy h betű, ha egy egész számot short típusként vagy egy l betű, ha long típusként írunk ki

Formátumozott kivitel

formátum karakter	típus
ld	long
d	int, short, char
f	float
c	char, int
Lg	long double

Matematikai függvények

A matematikai függvények a math.h nevű fejállományban vannak.

Állandók: PI értéke

M_PI 3.14159265358979323846

Függvények

```
double sin (double);
double cos (double);
double tan (double);
double sinh (double);
double cosh (double);
double tanh (double);
double asin (double);
double acos (double);
double atan (double);
double atan2 (double, double);
double exp (double);
double log (double);
double log10 (double);
double pow (double, double); double
sqrt (double);
```

```
double ceil (double); // Felfele
kerekít
double floor (double); // Felfele
kerekít
double fabs (double); // Abszolút
érték
double ldexp (double, int);
double frexp (double, int*);
double modf (double, double*);
double fmod (double, double);
int abs(int) //Abszolút érték
```

Véletlen szám

- Az stdlib.h fájlban találhatók a következő függvények.
  - int rand(void)
  - void srand(unsigned int)

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    printf("Véletlen\n");
    srand(time(NULL));
    printf("0 és 4 között: %d\n", rand() % 5);
    printf("20 és 29 között: %d\n", rand() % 10 +20);
    return 0;
}
```

Konverzió

```
double atof(const char*)
int atoi(const char*)
long atol(const char*)

int ltoa(int, char*, int)
```

Bevitel

Szám bekérése

```
#include <stdio.h>

main()
{
        int a;
        printf("Szam: ");
        scanf("%d", &a);
        printf("Ezt írtad: %d\n", a);
}
```

Szöveg bekérése

```
#include <stdio.h>

main()
{
        char nev[100];

        printf("Név: ");
        fgets(nev, 100, stdin);
        printf("Ezt írtad: %s\n", nev);
}
```

Egyszerű feladatok

- Írjunk olyan programot, amelyik szögmértékeket vált át fokról radiánra (és esetleg viszont)!
- Váltsuk át Celsius hőmérsékleti adatokat Fahrenheit fok egységbe! (F=1,8 * C +32)
- Számoljuk ki a kör területe, kerülete az átmérőjéből vagy a sugarából!
- Mit csinálhat a következő kódrészlet? Oldjuk meg a feladatot C segítségével!

```
be:a,b,c
l=(a + b > c && a + c > b && b + c > a)
ki:l
```

Szelekció: mondatszerű leírás

egyágú:

HA feltétel **AKKOR** utasítások
ELÁGAZÁS VÉGE

kétágú:

HA feltétel **AKKOR** utasítások **KÜLÖNBEN** utasítások
ELÁGAZÁS VÉGE

többágú

ELÁGAZÁS
felt1 **ESETÉN** utasítások
felt2 **ESETÉN** utasítások
…
feltN **ESETÉN** utasítások
feltN+1 **ESETÉN** utasítások
{**KÜLÖNBEN** utasítások}
ELÁGAZÁS VÉGE

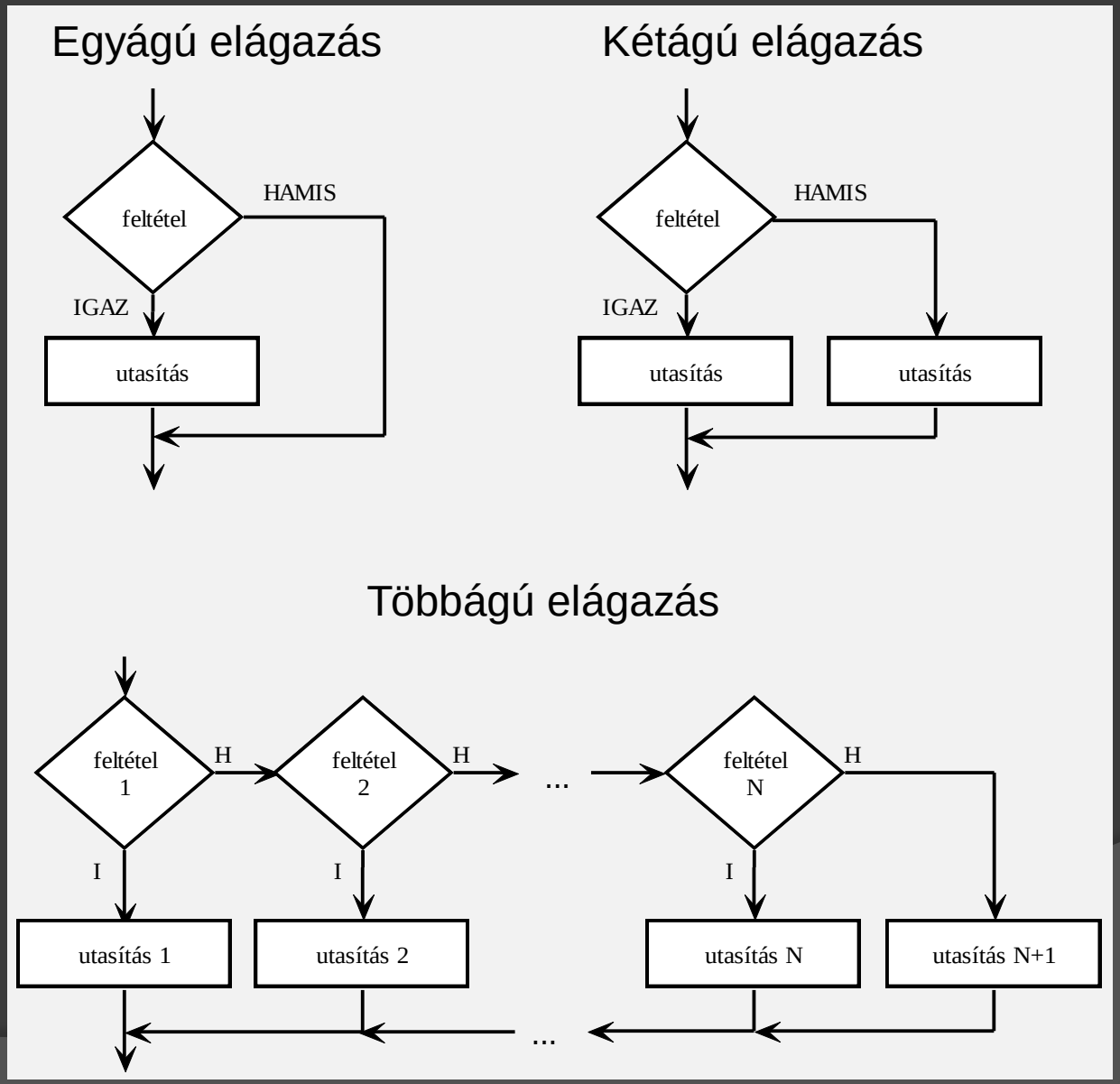
Mondatszerű példa

```
PROGRAM egyagu
   BE: szam
   HA szam>0 AKKOR KI: 'A szám pozitív'
   ELÁGAZÁS VÉGE
PROGRAM VÉGE
```

```
PROGRAM ketagu
   BE: szam
   HA szam<0  AKKOR KI: 'A szám pozitív'
              KÜLÖNBEN KI: 'A szám negatív'
   ELÁGAZÁS VÉGE
PROGRAM VÉGE
```

```
PROGRAM osztalyzat
   BE: jegy
   ELÁGAZÁS
      jegy = 5 ESETÉN KI: 'Jeles'
      jegy = 4 ESETÉN KI: 'Jó'
      jegy = 3 ESETÉN KI: 'Közepes'
      jegy = 2 ESETÉN KI: 'Elégséges'
      jegy = 1 ESETÉN KI: 'Elégtelen'
      KÜLÖNBEN KI: 'Hibás adat'
   ELÁGAZÁS VÉGE
```

Szelekció: folyamatábra



Szelekció

- if
- if-else
- switch

```c
#include <stdio.h>

main()
{
        int a = 5;
        if(a > 0)
                printf("Pozitív szám\n");

}
```

```c
#include <stdio.h>

main()
{
        int a = 5;
        if(a > 0)
                printf("Pozitív szám\n");
        else
                printf("Nulla vagy negatív\n");

}
```

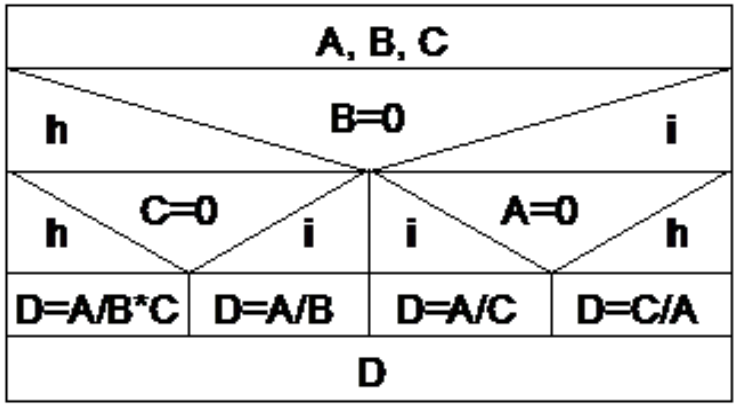
```c
#include <stdio.h>

main()
{
        int szam;

        printf("Szám: ");
        scanf("%d", &szam);

        switch(szam)
        {
                case 1 :
                        printf("Egy\n"); break;
                case 2 :
                        printf("Kettő\n"); break;
                default:
                        printf("Más szám\n");
```

Minden case részt meg kell szakítani egy break utasítással, ha nem szeretnénk a következő case utáni rész is végrehajtani.

Folyamatábra feladat

Gondoljuk végig mit csinál a program, s az eredményt „jósoljuk” meg!

```
a. A=100, B=10, C=5 ------> D=......
b. A=5, B=0, C=15 --------> D=......
c. A=10, B=1, C=2 --------> D=......
```



- Készítsük el a folyamatábráját!
- Programozzuk le C nyelv segítségével!

Feladat

Többágú feltételes utasítást mutatja be az alábbi mondatszerű megoldás. Valósítsuk ezt meg, majd gondoljuk át tudnánk-e „szebben” csinálni!

```
PROGRAM honapNapjainakSzama
    BE: sorszam
    ELÁGAZÁS
        sorszam=1 ESETÉN KI: ’31 napos!’
        sorszam=2 ESETÉN KI: ’28 napos (ha nem
szökőév!)’
        sorszam=3 ESETÉN KI: ’31 napos!’
        sorszam=4 ESETÉN KI: ’30 napos!’
        sorszam=5 ESETÉN KI: ’31 napos!’
        sorszam=6 ESETÉN KI: ’30 napos!’
        sorszam=7 ESETÉN KI: ’31 napos!’
        sorszam=8 ESETÉN KI: ’31 napos!’
        sorszam=9 ESETÉN KI: ’30 napos!’
        sorszam=10 ESETÉN KI: ’31 napos!’
        sorszam=11 ESETÉN KI: ’30 napos!’
        sorszam=12 ESETÉN KI: ’31 napos!’
        KÜLÖNBEN KI: ’Hibás adat!’
    ELÁGAZÁS VÉGE
PROGRAM VÉGE
```

Feladat

A testtömeg-indexet a következő képlet alapján számítható ki:

$$TTI = \frac{testsúly\,(kg)}{(testmagasság\,(m))^2}$$

Készítsünk olyan programot, ami bekéri a testsúly (kg), magasságot (cm) és kiszámolja a tti-t!

Ellenőrző kérdések

- Mi az az math.h?
- A maradék képzést milyen operátorral valósítjuk meg?
- Mondjon két egy operandusú operátort.
- Mondjon négy bitenkénti operátort.
- Milyen utasítással írathatunk ki formázott kivitelt.
- Formázott kivitel esetén, valós számot milyen formátumkarakterrel tudunk kiíratni?