

6. Laborgyakorlat

Számonkérés

Az alábbi példaprogram két beolvasott szám közül kiírja a nagyobbikat.

```
#include <stdio.h>

int main(void)
{
    double a, b;
    printf("Kerem az elso szamot!\n");
    scanf("%lf", &a);
    printf("Kerem a masodik szamot!\n");
    scanf("%lf", &b);
    if ( a > b )
        printf("%lf\n", a);
    else
        printf("%lf\n", b);
}
```

A következő program beolvas 2 számot, melyek egy téglalap oldalait jelölik (a és b), ez alapján kiszámolja a kerületet és a területet.

```
#include <stdio.h>

int main(void)
{
    double b=0,a=0;
    do
    {
        printf("Kerem az 'a' oldalt: ");
        scanf("%lf", &a);
    } while ( a<=0 );

    do
    {
        printf("Kerem a 'b' oldalt: ");
        scanf("%lf", &b);
    } while ( b<=0 );

    printf("A teglalap kerulete %lf\n",2*(a+b));
    printf("A teglalap terulete %lf\n",a*b);
    if (a==b)
        printf("A teglalap oldalai egyformak ezert negyzet...\n");
}
```

Az argumentumok kezelése

A program, az argumentumként megadott számokkal végez matematikai műveleteket

```
#include <stdio.h>
#include <math.h>

int main(int argc, char *argv[]) {
    printf("Argumentumok szama: %d\n",argc);
    printf("Az elso szam: %d\n",atoi(argv[1]));
    printf("A masodik szam: %d\n",atoi(argv[2]));
    printf("A ket szam szorzata: %d\n",atoi(argv[1])*atoi(argv[2]));
    printf("Az elso szam negyzete: %d\n",pow(atoi(argv[1])));
    printf("A masodik szam gyoke: %d\n",sqrt(atoi(argv[2])));
}
```

Fordítsuk le a programot proba néven. Mivel a math.h-t használjuk, a gcc-nek meg kell adni a -lm kapcsolót :

```
gcc -lm -o proba proba.c
```

Ezzel létrejön a proba nevű futtatható állomány.

```
./proba elso masodik
```

Ekkor az argumentumok száma 3 (első argumentum az argv[0]—./proba).

Argumentumként megadott számokból a legnagyobbat kiírja:

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

int main(int argc, char *argv[])
{
    int a,b,i;

    if (argc <= 1)
    {
        printf("Tul keves argumentum... \n");
        exit(1);
    }

    sscanf(argv[1],"%d",&a);
    b=a;
    for(i = 2; i <= argc - 1 ; i++)
    {
        sscanf(argv[i],"%d",&a);
        if( a>b ) b=a;
    }
    printf("A legnagyobb szam: %d \n",b);
}
```

Emlékeztetőül

Fordításhoz a gcc parancsot kell használnia, ha van math.h akkor -lm kapcsolóra is szükség van!
A programot elindítani a ./programnev utasítással lehet!

Feladat

A feladatok elkészítéséhez a lépjen be vagy diak vagy root felhasználóként a rendszerre, majd a /tmp könyvtárba!!

1. Írjon programot, mely bekéri egy kör sugarát, kiszámolja a kör területét ($K=2*r*PI$) és a területét ($T=r^2*PI$) ($PI=3.14$)! (6 pont)
 - Ha kisebb-egyenlő a kör sugara mint 0 akkor írjon ki hibát és lépjen ki a programból
2. Írjon programot, mely bekér két sugarat, majd 2 koordinátát ! A 2 sugár, egy origó középpontú körgyűrűt jelöl. A program döntse el, hogy a koordinátákkal megadott pont a két sugár által kijelölt körgyűrűre esik-e . A pont távolsága az origótól $d=\sqrt{pow(x,2)+pow(y,2)}$. Akkor van a pont körgyűrűn, ha $r1 \leq d \leq r2$ (12 pont)
 - Az első bekért szám a kisebbik sugár. Ha kisebb-egyenlő mint nulla hibaüzenettel lépjen ki a program a visszatérési érték 3 legyen (exit(3); ehhez stdlib headert be kell tölteni)!
 - A második bekért szám a nagyobbik kör sugara. Ha kisebb-egyenlő mint a kisebbik, akkor hibaüzenettel lépjen ki a program. A visszatérési érték 4 (exit(4);) legyen!
 - A harmadik bekért szám a pont X koordinátája.
 - A negyedik bekért szám a pont Y koordinátája.
3. Írjon programot, melynek argumentumként megadunk 5 számot (az argumentumokat át kell alakítani egész típusú számokká(!) az atoi() függvény segítségével. Az atoi függvény argumentuma egy string, mely szám karaktereket tartalmaz. Ezt a stringet alakítja egész típusú számmá.)! A tömbben lévő számokat rendezze növekvő sorrendbe! (12 pont)
 - Kezelje azt a hibát, ha a több vagy kevesebb mint 5 argumentum van. Nem kell ellenőrizni, hogy a kapott argumentumok valóban számok-e
 - A sorrendbe állításhoz, két egymásba ágyazott for ciklusra és feltételekre lesz szüksége.

Értékelés:

60% (18 pont) 2-es

70% (21 pont) 3-as

80% (24 pont) 4-es

90% (27 pont) 5-ös