

Tartalomjegyzék

- 1 Elmélet
 - ◆ 1.1 2013 előadás
 - ◆ 1.2 Függvények
 - ◆ 1.3 Tömbök
 - ◆ 1.4 Függvények
 - ◆ 1.5 Egy példa
- 2 Feladatok
 - ◆ 2.1 Üzemananyag
 - ◆ 2.2 Fizika
 - ◆ 2.3 Maximum keresés (már megint)
 - ◆ 2.4 Összeg
 - ◆ 2.5 Rendezés
 - ◆ 2.6 Sakktábla
 - ◆ 2.7 Mátrixok, 2 dimenziós tömbök
 - ◆ 2.8 Prímtényező keresés
 - ◆ 2.9 Gyakoriság, hisztogram

Elmélet

2013 előadás

- Mivel az előadás előtt járunk, itt a múlt év előadásának anyaga:
<http://wiki.math.bme.hu/view/Informatika2-2013/Eloadas>

Függvények

Tömbök

- Tömbök esetén "http://wiki.math.bme.hu/tömbösítve" http://wiki.math.bme.hu foglalkozunk a változók helyét a memóriában.
- Példa deklarációra:

```
int a[10];
```

- Példa inicializálásra:

```
int t[10]={6,8,-2,6,123,-8,3,4,2,1};
```

- A tömb elemei 0-tól indexelődnek.
- Deklaráláskor tehát **le kell rögzítenünk egy konstanssal a tömb méretét**
- Fontos, hogy a méretet csak konstanssal adhatjuk meg. Néhány fordító támogatja a tömb méretének változóval való megadását, de ezt általánosan nem alkalmazhatjuk.
- Ha fordítási időben nem ismert, mekkora tömbre lesz szükség, használjuk dinamikus tömböt. (ld. később)
- Mindig kötelező a tömbnek méretet adni.
- A tömb nem másolható az = operátorral, végig kell iterálnunk elemenként a két tömbön másolásakor:

- Lehetőségünk van többdimenziós tömbök deklarálására.
- Többdimenziós tömbök sorfolytonosan tárolódnak a memóriában, tehát a következő két inicializálás (mely egyben példa is) ekvivalens:

```
int t[3][4][2]={34,-5,3,20,12,5,-1,0,4,77,12,-3,-14,3,1,23,75,2,10,24,78,1,2,7};
```

```
int t[3][4][2]={{34,-5},{3,20},{12,5},{-1,0}},{4,77},{12,-3},{-14,3},{1,23}},{75,2},{10,24},{78,1,2,7}};
```

Függvények

- Programozásban a **függvény** egy nagyobb program forráskódjának része, mely adott feladatot hajt végre, és többször felhasználható anélkül, hogy a program kódjának több példányban is tartalmaznia kellene.
- Egy függvény mindig bemenő paraméterek egy érték-halmazát kapja meg és futása után valamilyen visszatérési értéket ad vissza. Itt most a hangsúly az **értéken** van mindkét esetben.
- C nyelvben egy függvénynek mindig **tetszőleges számú és típusú argumentuma, és egyetlen visszatérési értéke van.**
- Az általános felépítés mindig a következő?:

```
visszatérési-típus függvénynév (argumentumdeklarációk){
    deklarációk és utasítások
}
```

- A függvény **függvényfej** és **függvénytörzs** épül fel.
- A **függvényfej** megadja a függvény **típusát** (milyen típusú adatot ad vissza), a függvény **nevét**, valamint **paramétereinek típusát és nevét**.

```
int hatvanyozo(int alap, int kitevo)
```

- A **függvénytörzs** tartalmazza a függvényt felépítő utasítássorozatot.

```
{
    int i;
    for( ... )
    ...
    return ... ;
}
```

- A **függvénydefiníció** maga a függvény, azaz a függvényfej és a függvénytörzs együtt.

```
int hatvanyozo(int alap, int kitevo){
    int i;
    for( ... )
    ...
    return ... ;
}
```

- A függvény **prototípusa** a függvényfej pontos vesszővel lezárva, paraméternevek nélkül.

```
int hatvanyozo(int,int);
```

- A **függvénydeklaráció** a függvény prototípusa paraméterek nélkül. Csak a függvény típusát és nevét tartalmazza.

```
int hatvanyozo();
```

- A **prototípus** és **deklaráció** jelentősége, hogy egy C program csak olyan függvényt tud meghívni, amelynek deklarációja megel?zi a kódban a függvényhívást.

Egy példa

Tömböket és függvényeket tartalmazó példa.

```
#include<stdio.h>

void f(int a[], int N){
    int i = 0;
    for( i =0; i<N; i++){
        printf("%d ",a[i]);
    }
    printf("\n");
}

int main(void) {
    int a[2] = {1,2};
    f(a,2);
    return 0;
}
```

Feladatok

Üzemanyag

- A program bemenetként kapja meg, hogy a felhasználó hány gallon üzemanyagot használna fel.
- A program írja ki az alábbi információkat:
 - ◆ A felhasznált üzemanyag mennyisége literben
 - ◇ 1 gallon = 3.7854 l
 - ◆ Az üzemanyag gyártásához szükséges olaj mennyisége hordóban mérve.
 - ◇ Mindenképpen egész hordók vásárlására van szükség.
 - ◇ 1 hordó olajból 19.5 gallon üzemanyag készíthet?.
 - ◆ A megfelel? CO2 terhelés kg-ban
 - ◇ 1 gallon üzemanyag esetén ez ~ 9 kg CO2
 - ◆ A hordókban vásárolt alapanyag olaj ára az aktuális árfolyamon 3 különböz? pénzemben: EUR, USD, HUF.

Fizika

- A program bemenetként kapja meg, hogy az ?rutazáshoz használt ?rhajó sebessége hány %-a a fény sebességének (300 000 000 m/s).
 - ◆ Például 10% esetén a bemenet 0.1 és ekkor az ?rhajó sebessége 30 000 000 m/s
 - ◆ Az ?rhajó tömegét, feltéve, hogy nyugalmi tömege 70,000 kg.
 - ◆ Azon eltelt id?ket, melyeket az ?rhajósok érzékelnének, ha a következ? objektumokra utaznának a Földr?l:
 - ◇ Alpha Centauri: 4.3 fényév
 - ◇ Androméda Galaxis 2 000 000 fényév
 - ◆ Segítség:
 - ◇ <http://hu.wikipedia.org/wiki/Id%C5%91dilat%C3%A1ci%C3%B3>
 - ◇ http://hu.wikipedia.org/wiki/Relativisztikus_t%C3%B6meg

Maximum keresés (már megint)

- Írj függvényt, mely visszaadja egy tömb maximális elemét
- A függvény paraméterként kapja meg a tömböt és annak méretét
- Próbáld ki a függvényt úgy, hogy egy main függvényen belül hozol létre egy példa tömböt.

Összeg

- Írj függvényt, mely összeadja egy tömb elemeit.

Rendezés

- Írj függvényt, mely nagyság szerint rendezi egy tömb elemeit.

Sakktábla

- Rajzolj ki egy $N \times N$ -es sakktábla mintát, ahol X-szel jelöljük a fekete mezőket, és üresen hagyjuk (egy szóközzel) a fehéreket. Nem kell keretet adni a táblának.
- A sakktábla méretét (N) a felhasználótól kérd be.

Mátrixok, 2 dimenziós tömbök

- Írd ki egy 2 dimenziós tömb elemeit a képernyőre. Próbáld ki a függvényt úgy, hogy egy main függvényen belül hozol létre egy példa tömböt.
- Hasonlóan számold ki és jelenítsd meg a mátrix transzponáltját és négyzetét is.
- Írj függvényt, mely összead / kivon egymásból / összeszoroz 2 mátrixot és a megoldást megjeleníti a képernyőn.

Prímtényező keresés

- Írj programot, ami megkeresi egy a felhasználó által adott szám prímtényezőit és sorban kiírja azokat.
- Meg lehet oldani úgy is, hogy egyesével megpróbáljuk elosztani az adott számunkat 2-től kezdve egyesével haladva egész számokkal, amíg 1-hez nem jutunk.
- A maradék képzés (modulo) jele C-ben is a %

Gyakoriság, hisztogram

- Írj függvényt, mely megszámolja egy egész számokból álló tömb elemeinek a gyakoriságát.
- A függvény paraméterként kapja meg a vizsgált tömböt és annak méretét. Feltételezzük, hogy a tömbben 20-nál kisebb, vagy egyenlő értékek szerepelnek (miért van erre szükség?).