

Tartalomjegyzék

- 1 Típusok, tömbök, függvények
 - ◆ 1.1 Miért nem fut végtelen ciklusba a következ? kód?
 - ◆ 1.2 Adatlap
 - ◆ 1.3 Hatvány függvény
 - ◆ 1.4 Két szám összege
 - ◆ 1.5 Minimum
 - ◆ 1.6 Számláló
 - ◆ 1.7 Keres?
 - ◆ 1.8 Medián

Típusok, tömbök, függvények

A lenti feladatokat a CodeLite-ban oldhatjátok meg legegyszer?bben (ne felejtsetek el minden feladathoz új projektet nyitni!), de akinek a konzolos, gcc-s fordítás a szimpatikusabb úgy is csinálhatja.

Miért nem fut végtelen ciklusba a következ? kód?

```
#include<stdio.h>
int main(void){
    char i;
    for(i = 1; i != 0; i++){}
    return 0;
}
```

Módosítsd úgy a kódot, hogy minden lépésnél írja ki a képerny?re *i* értékét, szóközzel elválasztva! (Újsor jelet csak a ciklus után írj ki, egyszer!)

Adatlap

Kérj be adatokat a felhasználótól egy emberr?l, mégpedig a következ?ket:

- születési év
- magasság méterben
- tömeg kilogrammban

Majd írd ki a képerny?re ugyanezeket, szépen formázva, egymás alá rendezve, így nézzen ki (használd a printf-ben a "http://wiki.math.bme.hu\t"http://wiki.math.bme.hu-t arra hogy tab-ot írj ki):

```
Kora:      19      ev
Magassaga: 1.75    m
Tomege:    69      kg
```

Válaszd meg jól a változók típusait amiket használsz!

A tört érték kiírásának formázásához segítség: A printf-ben "http://wiki.math.bme.hu%1.8f"http://wiki.math.bme.hu -fel jelölheted hogy egy float típusú változót úgy akarsz kiírni hogy az egész részt egy karakterrel írod ki, a tört részt pedig 8 karakterrel.

Hatvány függvény

Írj hatvány függvényt ("http://wiki.math.bme.hu/hatvany" "http://wiki.math.bme.hu" névvel), első paramétere az alap, második a kitevő (ne felejtjük el a negatív kitevőket, de elég ha csak egész kitevőkkel működik). A main() függvényben hívd meg a "http://wiki.math.bme.hu/hatvany()" "http://wiki.math.bme.hu" függvényt néhányszor, és írd a képernyőre az eredményt.

Két szám összege

Írj egy függvényt, mely összead két egész számot, és kiírja az összeget, de a túlcordulást egy szöveges üzenettel jelzi. Többféle megoldás létezik. Ha az egyik szám pozitív ($y > 0$), akkor megnézheted, hogy a másik nagyobb-e, mint $INT_MAX - y$. Például használható a következő kód:

```
((y > 0 && x > INT_MAX - y) || (y < 0 && x < INT_MIN - y))
```

Minimum

Írj függvényt ami a beadott (double típusú elemeket tartalmazó) tömbnek visszaadja a legkisebb elemét (első paraméter a tömb, második a tömb mérete). Itt is (és a többi feladatnál is) a main() végezze az eredmény kiírását (és az esetleges beolvasást a felhasználótól az elején, de ez nem fontos, a main() függvény kódjában közvetlenül is definiálhatod az átadandó értékeket).

Számláló

Írj függvényt ami a beadott karaktereket tartalmazó tömbben megszámolja hányszor szerepel egy adott karakter (első paraméter a tömb, második a tömb mérete, harmadik a keresendő elem).

Keresés

Módosítsd úgy az előző függvényt, hogy a keresendő elem első fordulási indexét adja vissza.

Medián

Bemenet: maximum 30 pozitív egész szám, 0-val jelezzük ha vége a bemenetnek. Kimenet: a számsorozat mediánja (vagyis a középső elem ha sorbarendezzük a számokat).

A feladat megoldásához előbb nagyság szerint sorba kell rendezni a számokat. Erre többféle algoritmus létezik. Ha szereted a kihívásokat, akkor ki is találhatsz egyet, ha viszont nem, vagy nincs már idő a gyakorlat végén, akkor használd a buborék rendezést:

http://wiki.prog.hu/wiki/Bubor%C3%A9krendez%C3%A9s_%28algoritmus%29 . De ebben az esetben se copy-paste-elj, hanem értsd meg az algoritmust és gépeled be a saját változóneveiddel.

Legyen a main()-en kívül két függvény is a programodban, az egyik a rendezést oldja meg, a másik egy rendezett tömbből visszaadja a medián értékét.