

Gyakorló feladatsor C nyelvben

Svec Antal

2023.03.30

Adattípusok, operátorok, elágazások

1. Írjon C programot, amely bekér három pozitív számot, és eldönti, hogy lehetnek-e egy háromszög oldalai!
2. Írj programot, amely bekéri a felhasználó életkorát, majd kiírja, hogy nagykorú-e vagy sem!
3. Írj programot, amely bekéri a felhasználó testsúlyát és magasságát, majd kiszámolja a testtömegindexet (BMI) és kiírja az eredményt!
4. Írj programot, amely bekéri a felhasználó két egész számot és kiírja az összegüket, különbségüket, szorzatukat és hányadosukat!
5. Írj programot, amely bekéri a felhasználó egy egész számot és kiírja, hogy páros-e vagy páratlan!
6. Írj programot, amely bekéri a felhasználó két lebegőpontos számot és kiírja a nagyobbikat!
7. Írj programot, amely bekéri a felhasználó három egész számot és kiírja a legkisebbet!
8. Írj programot, amely bekéri a felhasználó két karaktert és kiírja, hogy azonosak-e! Karakter beolvasáshoz segítség:

```
char karakter;  
scanf("%c%c", &karakter);  
// %c formátum hozzáadásával csak egy darab karaktert olvas be és \n  
// karaktert figyelmen kívül hagyja. Ezáltal több karakter beolvasást  
// végezhetünk a scanf() függvény segítségével.
```

9. Írj programot, amely bekéri a felhasználó két lebegőpontos számot és kiírja az összegüket, különbségüket, szorzatukat és hányadosukat!
10. Írj programot, amely bekéri a felhasználótól egy kör átmérőjét és kiírja a hozzá tartozó kerületet és területet! (A pi értékét M_PI nevesített konstans a <math.h> könyvtárból használható. Program fordításkor a -lm kapcsolót írd végére. \$ gcc kor.c -lm)

Ciklusok, tömbök

1. Írj egy programot, amely kiírja az összes páros számot 1 és 100 között.
2. Írj egy programot, amely kiírja az összes osztóját a felhasználó által megadott egész számnak.
3. Írj egy programot, amely kiírja az összes prímszámot 1 és 100 között.
4. Írj egy programot, amely bekéri a felhasználó által megadott n pozitív egész számot, majd kiírja az első n darab Fibonacci-számot, amelyek párosak.

- Írj programot, ami bekér egy egész számot (**num**) a felhasználótól és utána addig kér be a felhasználótól új számokat, ameddig a számok összege nem haladja meg a **num** értékét. Írd ki a bekért számoknak az átlagát.
- Írj egy programot, amely bekéri a felhasználótól egy n pozitív egész számot, majd egyenként bekéri az n darab egész számot. Beolvasás után kiírja a megadott számok összegét, átlagát (lebegőpontos szám), legnagyobb és legkisebb számot.
- Írj egy programot, amely bekér egy pozitív egész számot (vizsgáljuk meg, hogy valóban pozitív számot adtak meg, ha nem akkor írjon ki egy hibaüzenetet és érjen végen a program), majd kirajzol egy háromszöget, amelynek minden sorában az adott sor sorszáma szerinti számok találhatók.

Példa: `$. /a.out *Adj meg egy egész pozitív számot : 5 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5*`

- Írj egy programot, amely bekér egy pozitív egész számot, majd kirajzol egy $n \times n$ méretű négyzetet a konzolra.

Példa: `$. /a.out`

```
Addj meg egy pozitív egész számot: 5
*****
*   *
*   *
*   *
*****
```

Sztringek, függvények eljárások

- Írj egy programot, amely beolvas egy sztringet a felhasználótól, majd megfordítja azt és kiírja az eredményt a konzolra!
- Írj egy programot, amely beolvas egy sztringet a felhasználótól, majd megszámlálja az összes szóközt az adott sztringben és kiírja az eredményt a konzolra! A szóközek számolásához írd meg egy függvényt, ami paraméterként fogad egy stringet és visszatérítési értéként a szóközök számát adja vissza.
- Írj egy programot, amely beolvas egy sztringet a felhasználótól. Majd írd meg egy függvényt, ami ellenőrzi, hogy az adott sztring tartalmaz-e legalább egy nagybetűt, egy kisbetűt, egy számot és egy speciális karaktert (pl. !, #, \$, % stb.). Ha igen akkor a függvény térjen vissza 1-el, különben 0-val. legalább 5 példa segítségével teszteld a működést.

- Írj egy programot, amely beolvas két sztringet a felhasználótól, majd összefűzi azokat és kiírja az eredményt a konzolra. Ezt egy eljárás segítségével old meg, amelynek definíciója a következő:

```
void osszefuz(char dest[], char str1[], str2[]);
```

A dest változóba legyen összefűzve a str1 és str2 változóban lévő stringek. Ügyelj arra, hogy a dest karakter tömb mérete elegendő legyen, hogy a két string beleférjen. Egy kis segítség:

```
char str1[64];
char str2[64];
char osszefuzott[128];
//beolvasas
osszefuz(osszefuzott, str1, str2);
printf("%s + %s = %s", str1, str2, osszefuzott);
/*
Példa:
str1 = "Hello"
str2 = "world"
Kimenet:
Hello + world = Helloworld
*/
```

- Írj programot, ami minden sorba kiírja az angol abc nagy betűjét és mellette a nagy változatát.

Példa: *\$/a.out*

```
A a
B b
C c
...
Z z
```

Parancssori argumentumok

- Írj programot ami addig kér be a felhasználótól szavakat ameddig a felhasználó nem írja be egy parancssori argumentumban megadott szót.

Példa: *\$/a.out kaposztaleves*

```
Írj be szavakat:
: abc
: sfasdf
: kaposztaleves
Vége!
```

- Írj egy programot, amely fogad csak egy egész számot a parancssori argumentumokon keresztül, majd kiszámolja és kiírja az adott szám faktoriálisát! Legfeljebb a 12-es értékkel próbálkozz.
- Írj egy programot, amely fogad egy parancssori argumentumként szavakat, vesszővel elválasztva, majd szavanként végigmegy rajta, és kiírja az összes szót kisbetűkkel!