

Vezérlési szerkezetek II. - ciklusok (for, while, do..while)

16. Készítsünk programot, amely billentyűzetről bekéri a felhasználó nevét egy **char nev[30];** változóba, majd kiírja a képernyőre a beolvasott nevet 100-szor, szóközzel elválasztva.

A program kimenete:

Neved: **Snoopy**

[illegible]

17. Készítsünk programot, amely kiszámolja az első n drb. természetes szám összegét, majd kiírja az eredményt.

Az összeg kiszámolásához vezessünk be egy változót, amelyet a program elején kinullázunk, a ciklusmagban (a kapcsos zárójelek között) pedig mindig hozzáadjuk a ciklusváltozó értékét, tehát sorban az **1, 2, 3, ..., n** számokat.

A program kimenete:

N = 26

Az első 26 szám összege: 351

18. Készítsünk programot, amely kiszámolja az első n drb. természetes szám szorzatát egy ciklus segítségével.

A szorzat kiszámolásához vezessünk be egy változót, amelyet a program elején beállítunk **1-re**, a ciklusmagban (a **kapcsos zárójelek között**) pedig mindig hozzászorozzuk a ciklusváltozó értékét, tehát sorban az **1, 2, 3, ..., n** számokat.

A program kimenete:

N = 7

Az elso 7 szam szorzata: 5040

19. Készítsünk programot, amely kiszámolja az első n db. páros szám összegét.

A ciklusváltozót vegyük **1-től n-ig**, majd a ciklusmagban (a kapcsos zárójelek között) vegyük a ciklusváltozó kétszeresét (**$i*2$**). Így megkapjuk a páros számokat, melyeket hasonlóan adjunk össze, mint a 17. feladatban.

A program kimenete:

N = 15

Az első 15 drb. páros szám összege: 240

20. Készítsünk programot, amely kiszámolja az első **n** drb. páratlan szám összegét.

A ciklusváltozót vegyük **1-től n-ig**, majd a ciklusmagban (a kapcsos zárójelek között) vegyük a ciklusváltozó kétszeresét eggyel csökkentve (**$i*2-1$**). Így megkapjuk a páratlan számokat, melyeket hasonlóan adjunk össze, mint az előző feladatban.

A program kimenete:

N = 15

Az első 15 drb. páratlan szám összege: 225

21. Készítsünk programot, amely beolvas egy egész számot, majd kiírja a beolvasott számtól **0-ig** az egész számokat a képernyőre. A léptetést **$i++$** helyett **$i--$** kifejezéssel végezzük el!

A program kimenete:

N = 20

20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

22. Kérjünk be két természetes számot (**m, n**), majd rajzoljunk ki a képernyőre egy **m** szélességű és **n** magasságú téglalapot csillag (*) jelekből.

A program kimenete:

A téglalap szélessége: **10**

A téglalap magassága: **4**

```
*****
*****
*****
*****
```

23. Határozzuk meg és írassuk ki az összes hárommal vagy öttel osztható, **200-nál** kisebb természetes számot.

A program kimenete:

3, 5, 6, 9, 10, 12, 15, 18, 20, 21, 24, 25, 27, 30, 33, 35, 36, 39,
40, 42, 45, 48, 50, 51, 54, 55, 57, 60, 63, 65, 66, 69, 70, 72, 75,
78, 80, 81, 84, 85, 87, 90, 93, 95, 96, 99, 100, 102, 105, 108,
110, 111, 114, 115, 117, 120, 123, 125, 126, 129, 130, 132, 135,

138, 140, 141, 144, 145, 147, 150, 153, 155, 156, 159, 160, 162,
165, 168, 170, 171, 174, 175, 177, 180, 183, 185, 186, 189, 190,
192, 195, 198,

24. Készítsünk programot, amely beolvas egy n természetes számot, majd billentyűzetről bekér n db. természetes számot és ezeket a számokat összeadja, majd kiírja az eredményt.

Vegyünk egy változót, amit a program elején kinullázunk. Ehhez a cikluson belül mindig adjuk hozzá az éppen beolvasott számot. A szám beolvasása a ciklusban lehet n -szer ugyanabba a változóba, hiszen miután hozzáadtuk az összeghez, már nincs rá szükségünk, tehát használhatjuk a következő szám beolvasására.

A program kimenete:

Mennyi számot szeretnél megadni? $n=7$

1. szám: 18
2. szám: 21
3. szám: 11
4. szám: 6
5. szám: 91
6. szám: 85
7. szám: 27

A megadott számok összege: 259

25. Készítsünk programot, amely bekér egy k pozitív egész számot, majd kiszámolja a következő összeget: $1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + 4 \cdot 5 + \dots + k \cdot (k+1)$

A program kimenete:

$k = 10$
 $1 \cdot 2 + 2 \cdot 3 + \dots + k \cdot (k+1) = 440$

26. Kérjünk be egy n természetes számot, majd írassuk ki a három összes olyan többszörösét, amely kisebb vagy egyenlő mint n .

A program kimenete:

$n = 22$
3, 6, 9, 12, 15, 18, 21,

27. Kérjünk be két természetes számot (m, n), majd rajzoljunk ki a képernyőre egy paralelogrammát csillag (*) jelekből (a paralelogrammának n sora legyen, mindegyik sorban m csillaggal).

A program kimenete:

A paralelogramma alapjának hossza: 12
A paralelogramma magassága: 5

```

*****
*****
*****
*****
*****
*****

```

28. Kérjünk be egy természetes számot (**a**), majd rajzoljunk ki a képernyőre egy háromszöget csillagokból (*). A háromszög **a** sornyi csillagból álljon.

A program kimenete:

A haromszog magassaga: 5

```

*
**
***
****
*****
*****

```

29. Kérjünk be két természetes számot (**m**, **n**), majd rajzoljunk ki a képernyőre egy **m** szélességű és **n** magasságú téglalapot csillag (*) jelekből úgy, hogy a téglalap belseje üres legyen.

A program kimenete:

A teglalap szelessege: 10
A teglalap magassaga: 5

```

*****
*           *
*           *
*           *
*           *
*****

```

30. Állítsuk elő és írassuk ki az első **n** darab Fibonacci-számot. Ennek a sorozatnak az a jellemzője, hogy bármelyik eleme egyenlő az előző kettő összegével. A sorozat néhány eleme: **0, 1, 1, 2, 3, 5, 8, 13, ...**

A program kimenete:

n = 10
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

31. Állítsuk elő és írassuk ki az első **n** darab Fibonacci-szám összegét. Az összeg, melyet számoljon ki a program az első **n** elemből: **0 + 1 + 1 + 2 + 3 + 5 + 8 + 13 + ...**

A program kimenete:

n = 10
Az elso 10 Fibonacci szam osszege: 88

32. Készítsünk programot, amely egy **while** ciklus segítségével kiszámolja az első **n** szám összegét!

A program kimenete:

Kerek egy számot, $n = 7$
Az első 7 szám összege: 28

33. Készítsünk programot, amely beolvas egy egész számot, majd elosztja **2**-vel annyiszor, ahányszor lehet és közben felírja a számot kettes számok szorzataként megszorozva egy olyan számmal, amely már nem osztható **2**-vel.

A program kimenete:

Kerek egy számot: **128**
 $128 = 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 1$

A program kimenete:

Kerek egy számot: **72**
 $72 = 2 * 2 * 2 * 9$

A program kimenete:

Kerek egy számot: **15**
 $15 = 15$

34. Készítsünk programot, amely a felhasználótól bekér egész számokat mindaddig, amíg nem ad meg **0**-t. Számoljuk ki a megadott számok szorzatát, de ügyeljünk arra, hogy a **0**-t ne számoljuk bele!

A program kimenete:

Kerek egy számot (0 = a bevitel vege): **5**
Kerek egy számot (0 = a bevitel vege): **4**
Kerek egy számot (0 = a bevitel vege): **2**
Kerek egy számot (0 = a bevitel vege): **7**
Kerek egy számot (0 = a bevitel vege): **0**

A számok szorzata: 280

35. Készítsünk programot, amely a felhasználótól bekér egész számokat mindaddig, amíg nem ad meg **0**-t. Írjuk ki, hogy a felhasználó összesen mennyi számot adott meg és számoljuk ki a megadott számok átlagát.

Ügyeljünk arra, hogy a program a **0**-t ne számítsa bele a beadott számok darabszámba és az átlagába se, az csak a bevitel végét jelezze.

A program kimenete:

Kerek egy számot (0 = a bevitel vege): **2**
Kerek egy számot (0 = a bevitel vege): **7**
Kerek egy számot (0 = a bevitel vege): **5**

Kerek egy számot (0 = a bevitel vege): 0

A számok darabszáma: 3

A számok atlaga: 4.667

36. Készítsünk programot, amely bekér a felhasználótól egész számokat mindaddig, amíg a felhasználó nem ad meg 0-t. Közben határozzuk meg a megadott számok közül a legnagyobb számot. Ügyeljünk arra, hogy az utolsónak megadott 0-t ne vegyük figyelembe a maximum meghatározásánál!

A program kimenete:

Kerek egy számot (0 = a bevitel vege): 5
Kerek egy számot (0 = a bevitel vege): 18
Kerek egy számot (0 = a bevitel vege): 12
Kerek egy számot (0 = a bevitel vege): 21
Kerek egy számot (0 = a bevitel vege): 17
Kerek egy számot (0 = a bevitel vege): 0

A legnagyobb szám: 21

37. Készítsünk játékprogramot, amely gondol egy számot 1 és 50 között. A felhasználó addig találgathat, amíg nem találja el a gondolt számot. A számítógép minden rossz tipp után írja ki, hogy a gondolt szám nagyobb vagy kisebb.

1 és 50 közötti véletlen számot az alábbi módon tudunk kigenerálni (az **srand()** függvény inicializálja a véletlenszám generátort, a **rand()** függvény kigenerál egy egész számot):

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

main() {
    srand(time(NULL));
    int gondoltSzam = rand() % 50 + 1;

}
```

A program kimenete:

Gondoltam egy számra 1-50 között. Melyik ez a szám?

Tipped: 25

Kisebb számra gondoltam!

Tipped: 12

Kisebb számra gondoltam!

Tipped: 6

Nagyobb számra gondoltam!

Tipped: 9
Kisebb számra gondoltam!
Tipped: 8
Eltalaltad!

38. Egészítsük ki az előző programunkat úgy, hogy a játékos csak maximum 7-szer tippelhessen!
Ha a hetedik tippre sem találja el a gondolt számot, a program írja ki a gondolt számot majd fejeződjön be.

A program kimenete:

Gondoltam egy számra 1-50 között. Melyik ez a szám?
1. tipped: 25
Nagyobb számra gondoltam!
2. tipped: 37
Kisebb számra gondoltam!
3. tipped: 31
Nagyobb számra gondoltam!
4. tipped: 34
Kisebb számra gondoltam!
5. tipped: 33
Kisebb számra gondoltam!
6. tipped: 32
Eltalaltad!

A program kimenete:

Gondoltam egy számra 1-50 között. Melyik ez a szám?
1. tipped: 5
Nagyobb számra gondoltam!
2. tipped: 10
Nagyobb számra gondoltam!
3. tipped: 15
Nagyobb számra gondoltam!
4. tipped: 20
Nagyobb számra gondoltam!
5. tipped: 25
Nagyobb számra gondoltam!
6. tipped: 30
Nagyobb számra gondoltam!
7. tipped: 35
Kisebb számra gondoltam!
Nincs több lehetőséged tippelni!
A gondolt szám: 34

39. Készítsünk programot, amely ki fogja kérdezni a tízes szorzótáblát (az <1,10> intervallumból)!
A két számot a számítógép véletlenszám generátor segítségével válassza ki. A program akkor fejeződjön be, ha a felhasználó 10 példát kiszámolt helyesen. Rossz válasz esetén kérdezze újra ugyanazt a példát.

A program kimenete:

```
1. feladat: 10 * 8 = 80
2. feladat: 3 * 3 = 9
3. feladat: 1 * 9 = 1
3. feladat: 1 * 9 = 1
3. feladat: 1 * 9 = 1
3. feladat: 1 * 9 = 9
4. feladat: 3 * 8 = 24
5. feladat: 2 * 9 = 18
6. feladat: 6 * 7 = 42
7. feladat: 10 * 8 = 80
8. feladat: 5 * 3 = 15
9. feladat: 2 * 3 = 6
10. feladat: 7 * 10 = 70
```

40. Egészítsük ki a programunk úgy, hogy a végén írja ki az eredményességet százalékban!

Mivel a % jelnek speciális jelentése van a **printf()** függvény formátumsztringjében, ezért azt %% használatával tudjuk kiírni, pl.: **printf("Az elert eredmény 82 %%, ami B osztalyzatnak felel meg.");**

A program kimenete:

```
1. feladat: 8 * 4 = 32
2. feladat: 6 * 6 = 36
3. feladat: 8 * 8 = 64
4. feladat: 1 * 7 = 7
5. feladat: 5 * 1 = 5
6. feladat: 6 * 2 = 12
7. feladat: 3 * 8 = 24
8. feladat: 8 * 1 = 8
9. feladat: 2 * 5 = 1
9. feladat: 2 * 5 = 1
9. feladat: 2 * 5 = 1
9. feladat: 2 * 5 = 10
10. feladat: 6 * 10 = 60
```

Osszesen 3 valasz volt helytelen a 13-bol.
Eredmenyesseg: 76.92 %

41. Készítsünk programot, amely bekér egy egész számot és felbontja prímtényezőik szorzatára!

A feladat megoldásához előbb vizsgáljuk meg, hogy a beadott szám osztható-e **2**-vel. Amíg osztható, osszuk el **2**-vel és közben mindig írjuk ki a képernyőre a **2**-es prímtényezőt. Ha a szám már nem osztható **2**-vel, hasonlóan vizsgáljuk meg, hogy osztható-e többször is **3**-mal, majd **4**-gyel, **5**-tel, **6**-tal, **7**-tel, stb. mindaddig, amíg az eredeti szám változójának értéke nem lesz az osztások után **1**.

Bár a feladat megoldható különböző képpen, a legegyszerűbb megoldáshoz valószínűleg szükség lesz az utótesztelő ciklus (**do... while...**), az előtesztelő ciklus (**while...**) és a feltételvizsgálat (**if..**) használatára is a programban.

A program kimenete:

Kerek egy 1-nél nagyobb számot: **20216**

$20216 = 2 * 2 * 2 * 7 * 19 * 19$

A program kimenete:

Kerek egy 1-nél nagyobb számot: **20**

$20 = 2 * 2 * 5$

A program kimenete:

Kerek egy 1-nél nagyobb számot: **17**

$17 = 17$