



[Irányítópult](#) / [Kurzusaim](#) / [Programozás \(GKNB_INTM021\)](#) / [Vizsgák](#) / [2020. június 18. 13:15 \(nappali tagozat\)](#)

1 kérdés
Nincs befejezve
1 pont
szerezhető

Készítsen **binTizesbe** függvényt, amely stringet vár bemenetként és decimális számot ad vissza. A bemeneti string egy bináris számsor, ennek megfelelő számot adjon vissza a függvény.

Szintaktikai hiba esetén a fordító által jelzett sor számából vonjon ki 10-et, hogy megkapja a szerkesztőben olvasható sorszámot!

For example:

Test	Result
cout << binTizesbe("00000001") << "\n";	00000001 -- 1
cout << binTizesbe("00000010") << "\n";	00000010 -- 2
cout << binTizesbe("00000100") << "\n";	00000100 -- 4
cout << binTizesbe("00000101") << "\n";	00000101 -- 5
cout << binTizesbe("10000001") << "\n";	10000001 -- 129

Answer: (penalty regime: 0 %)

Reset answer

```
1 int binTizesbe(string n)
2 {
3     int decSzam = 0;
4     decSzam = decSzam + int(n);
5
6     cout << n << " -- ";
7     return -1;
8 }
```

Ellenőrzés

[◀ Mintavizsga](#)

Ugrás...

[Előadás fóliák ▶](#)

2 kérdés

Nincs befejezve

1 pont

szerezhető

Adott a következő struktúra, amely országot, az ország Nobel-dijasainak számát és a lakosságot (millió főben) tartalmazza:

```
struct NobelDij{
    string country;
    unsigned int numWinnners;
    double millionPeople;
}
```

Írjon meg két függvényt:

- `void sortByRate(bool rate, NobelDij *nobels, int n)` - a rendezést valósítja meg, ha `rate true`, akkor Nobel díj / millió lakos, ha `false` akkor pedig a Nobel díj alapján csökkenő sorrendbe
- `void statistics(NobelDij *nobels, int n)` - írja ki orszagonként átlagosan hány millió lakos és átlagosan mennyi a Nobel-díjas szám

Szintaktikai hiba esetén a fordító által jelzett sor számából vonjon ki 40-at, hogy megkapja a szerkesztőben olvasható sorszámot!

For example:

Test	Result
<pre>NobelDij nobels[] = { {"AUS", 12, 24.4}, {"BEL", 10, 11.42}, {"CZE", 5, 10.6}, {"DNK", 13, 5.73}, {"FRA", 68, 64.97}, {"GRE", 2, 11.1}, {"HUN", 13, 9.7}, {"GER", 107, 82.11}, {"SPA", 8, 46.35}, {"ROM", 4, 19.67}, {"CHE", 26, 8.47}, {"SLN", 1, 2.}, {"SLK", 0, 5.45}, {"USA", 368, 324.45}, {"UKR", 2, 44.222}, }; sortByRate(true, nobels, 15); printAll(nobels, 15); statistics(nobels, 15);</pre>	<pre>1. hely: CHE (3.07 Nobel díj / millio lakos) [Osszesen 26 Nobel] 2. hely: DNK (2.27 Nobel díj / millio lakos) [Osszesen 13 Nobel] 3. hely: HUN (1.34 Nobel díj / millio lakos) [Osszesen 13 Nobel] 4. hely: GER (1.30 Nobel díj / millio lakos) [Osszesen 107 Nobel] 5. hely: USA (1.13 Nobel díj / millio lakos) [Osszesen 368 Nobel] 6. hely: FRA (1.05 Nobel díj / millio lakos) [Osszesen 68 Nobel] 7. hely: BEL (0.88 Nobel díj / millio lakos) [Osszesen 10 Nobel] 8. hely: SLN (0.50 Nobel díj / millio lakos) [Osszesen 1 Nobel] 9. hely: AUS (0.49 Nobel díj / millio lakos) [Osszesen 12 Nobel] 10. hely: CZE (0.47 Nobel díj / millio lakos) [Osszesen 5 Nobel] 11. hely: ROM (0.20 Nobel díj / millio lakos) [Osszesen 4 Nobel] 12. hely: GRE (0.18 Nobel díj / millio lakos) [Osszesen 2 Nobel] 13. hely: SPA (0.17 Nobel díj / millio lakos) [Osszesen 8 Nobel] 14. hely: UKR (0.05 Nobel díj / millio lakos) [Osszesen 2 Nobel] 15. hely: SLK (0.00 Nobel díj / millio lakos) [Osszesen 0 Nobel] Orszagonkent atlagosan: 44.71 millio lakos Atlagos Nobel-dijas szam orszagonkent: 2.84</pre>

Answer: (penalty regime: 0 %)

Reset answer

```
1 void sortByRate(bool rate, NobelDij *nobels, int n)
2 {
3
4 }
5
6
7 void statistics(NobelDij *nobels, int n)
8 {
9
10 }
11
12
13 void printAll(NobelDij *nobels, int n)
14 {
15     for (int i = 0; i < n; i++)
16     {
17         cout << i+1 << ". hely: " << nobels[i].country << " (" << fixed << setprecision(2) << nobels[i].millionPeople << "
18     }
19 }
```

Ellenőrzés

3 kérdés

Nincs befejezve

1 pont

szerezhető

Írjon függvényt, ami egy 8x8-as sakktáblán fogadja el a ló lépéseit. A függvény neve legyen `lepesEllenor`. Az argumentumok: `bolS` jelölje a sort ahonnan, a `bol0` pedig az oszlopot ahonnan indul a lépés. A `baS` a sor, a `ba0` pedig az oszlop ahova érkezik a ló. Térjen vissza `true`-val, ha szabályos a lépés `false`-al, minden más esetben (szabálytalan lépés, hibás input).

Segítség:

	0	1	2	3	4	5	6	7
0								
1			A		B			
2		H				C		
3				X				
4		G				D		
5			F		E			
6								
7								

Szintaktikai hiba esetén a fordító által jelzett sor számából vonjon ki 10-et, hogy megkapja a szerkesztőben olvasható sorszámot!

For example:

Test	Result
cout << (lepesEllenor(3,3,2,1) ? " helyes lepes" : " hibas lepes") << "\n";	[3,3] -> [2,1] helyes lepes
cout << (lepesEllenor(3,3,1,2) ? " helyes lepes" : " hibas lepes") << "\n";	[3,3] -> [1,2] helyes lepes
cout << (lepesEllenor(3,3,4,1) ? " helyes lepes" : " hibas lepes") << "\n";	[3,3] -> [4,1] helyes lepes
cout << (lepesEllenor(3,3,5,2) ? " helyes lepes" : " hibas lepes") << "\n";	[3,3] -> [5,2] helyes lepes
cout << (lepesEllenor(3,3,3,4) ? " helyes lepes" : " hibas lepes") << "\n";	[3,3] -> [3,4] hibas lepes
cout << (lepesEllenor(9,9,8,7) ? " helyes lepes" : " hibas lepes") << "\n";	[9,9] -> [8,7] helyes lepes

Answer: (penalty regime: 0 %)

Reset answer

```
1 bool lepesEllenor(unsigned short bolS, unsigned short bol0, unsigned short baS, unsigned short ba0)
2 {
3     cout << "[" << bolS << "," << bol0 << "]" -> [" << baS << "," << ba0 << "];
4     return false; // ezt átírni
5 }
```

Ellenőrzés

4 kérdés

Nincs befejezve

1 pont
szerezhető

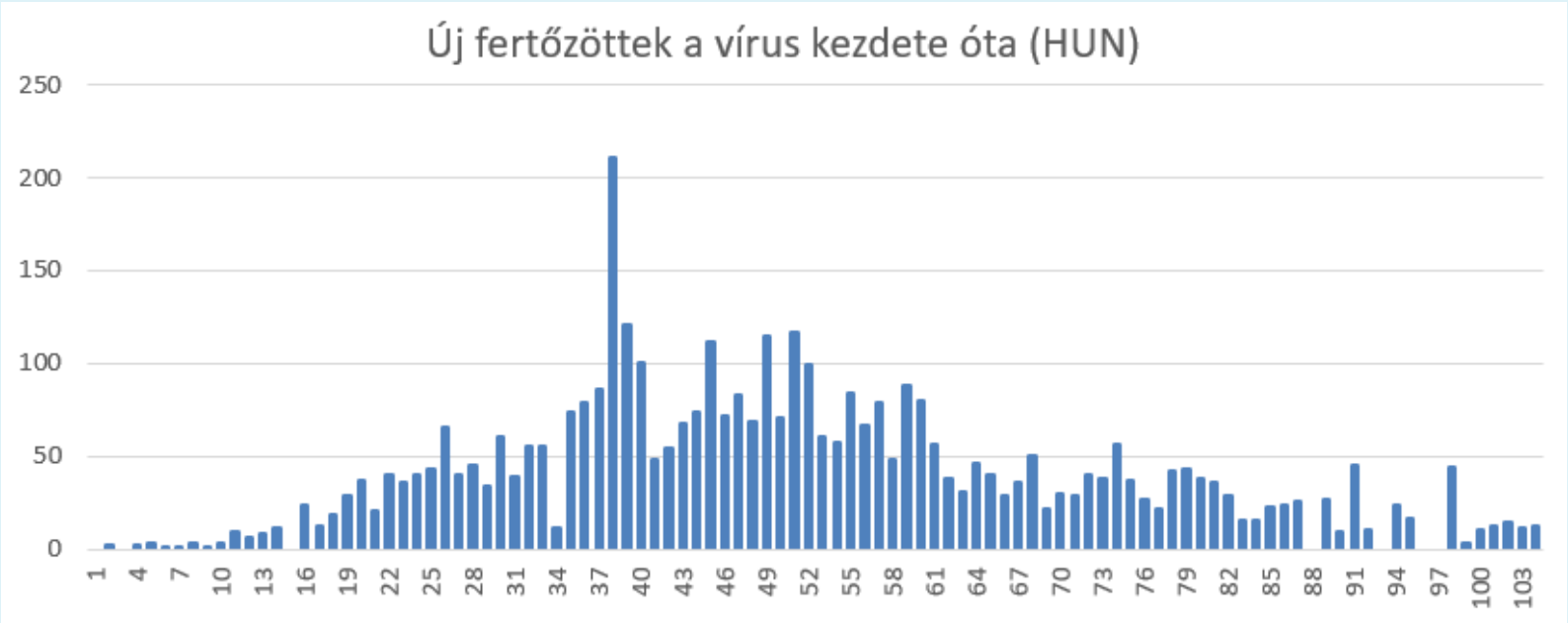
Készítsen függvényeket, amelyekkel fájlból olvas be egész számokat és azokat egy tömbbe illeszti. A számok a Magyarországon regisztrált COVID fertőzötteket jelentik a vírus megjelenésétől kezdve.

Valósítsa meg a 3 rövid függvényt:

- `void fajbolOlvas(string fajlnev, int adatok[], int db)` - Beolvassa *fajlnev*-ből az *adatok*-ba az *n* darabszámú adatot. A *fajlnev* nevű fájlnak legfeljebb az első *n* sorát, és az abban található adatokat helyezze el az *adatok* tömbben. Tesztesetként csak olyan eseteket adunk, ahol a fájl nagyobb vagy egyenlő *n*.
- `double atlag(const int adatok[], int db)` - Visszaadja a tömb átlag értékét.
- `double osszes(const int adatok[], int db)` - Visszaadja a tömb összértékét.

Például a `covid19.txt` tartalma:

2
0
2
3
1
1
3
1
3
9
6
8
11
0
23
12
....stb



Szintaktikai hiba esetén a fordító által jelzett sor számából vonjon ki 20-at, hogy megkapja a szerkesztőben olvasható sorszámot!

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2 #include <fstream>
3 #define FILE "covid19.txt"
4
5 // fájlból olvas
6 void fajbolOlvas(string fajlnev, int adatok[], int db)
7 {
8     ifstream file;
9     file.open(fajlnev);
10
11
12     file.close();
13 }
14
15 double atlag(const int adatok[], int db)
16 {
17     return -1;
18 }
19
20 double osszes(const int adatok[], int db)
21 {
```



[Irányítópult](#) / [Kurzusaim](#) / [Programozás \(GKNB_INTM021\)](#) / [Vizsgák](#) / [2020. június 18. 13:15 \(nappali tagozat\)](#)

5 kérdés

Nincs befejezve

1 pont
szerezhető

Írjon egy `maxoszlop` nevű függvényt, ami paraméterként kap egy dinamikusan, soronként foglalt int mátrixot (a sorok kezdőcímeinek tömbjét), a sorok számát, és oszlopok számát; és visszaadja annak az oszlopnak az indexét, amelyikben az elemek összege a legnagyobb. Ha több maximális összegű oszlop is van, ezek közül a legnagyobb indexű oszlop indexét adja vissza.

Megjegyzés: A tesztekben található létrehoz, feltölt, és felszabadít függvényeket nem kell megírnia.

Szintaktikai hiba esetén a fordító által jelzett sor számából vonjon ki 40-et, hogy megkapja a szerkesztőben olvasható sorszámot!

For example:

Test	Result
<pre>int sorok = 4, oszlopok = 5; int **mtx = létrehoz(sorok, oszlopok); feltolt(mtx, sorok, oszlopok, 0); cout << "A legnagyobb osszegu oszlop indexe: " << maxoszlop(mtx, sorok, oszlopok) << "\n"; felszabadit(mtx, sorok);</pre>	A legnagyobb osszegu oszlop indexe: 2

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <climits>
2
3 int maxoszlop(int **mtx, int s, int o) {
4
5     return -1; //
6 }
```

Ellenőrzés

[◀ Mintavizsga](#)

Ugrás...

[Előadás fóliák ▶](#)