

## Tíz kérdés (amelyből 3–5-re kell majd válaszolnod laboron, papíron)

1. Mi a `for` utasítás szintaxisa? Adj példát!
2. Mi a `while` utasítás szintaxisa? Adj példát!
3. Mi a `do-while` utasítás szintaxisa? Adj példát!
4. Mi egy alapvető különbség a `while` és `do-while` ciklusok között?
5. Milyen esetekben, és mire használható a `break` utasítás?
6. Hogyan működik a `continue` utasítás?
7. Egészíts ki a programrészletet úgy, hogy az 1,2,3,4,5,6,7,8,9,10 értékek jelenjenek meg a képernyőn! `for ( ..... ; ..... ; ..... ) { cout << ..... ; }`
8. Egészíts ki a programrészletet úgy, hogy az 5,5,5,5,5,5,5,5,5 értékek jelenjenek meg a képernyőn! `for ( ..... ; ..... ; ..... ) { cout << ..... ; }`
9. Egészíts ki a programrészletet úgy, hogy az 9,8,7,6,5,4,3,2,1,0 értékek jelenjenek meg a képernyőn! `for ( ..... ; ..... ; ..... ) { cout << ..... ; }`
10. Adj példát végtelen ciklusra!

## LABOR–feladatok (Debugging!!!)

1. Futtasd lépésenként az alábbi programot a tanártól kapott bemenetre! Értelmezd az eredményt!

```
#include <iostream>

using namespace std;

int main(){
    int i,n,x,s=0;

    cin >> n;

    for ( i=1 ; i<=n ; ++i ){
        cin >> x; s += x;
    }

    cout << "s= " << s;

    return 0;
}
```

2. Futtasd lépésenként az alábbi programot a tanártól kapott bemenetre! Értelmezd az eredményt!

```
#include <iostream>

using namespace std;

int main(){
    int x,p=1;

    cin >> x;

    while ( x>0 ){
        p *= x%10; x /= 10;
    }
}
```

```

        cout << "p= " << p;

        return 0;
    }

```

3. Futtasd lépésenként az alábbi programot a tanártól kapott bemenetre! Értelmezd az eredményt!

```

#include <iostream>

using namespace std;

int main(){
    unsigned i;
    double szam, osszeg = 0;

    for ( i=0 ; ; ++i ){
        cin >> szam;
        if ( !szam ) { break; }
        osszeg += szam;
    }

    if (i){ cout << "Atlag = " << osszeg / i;}
    else { cout << "Nem volt szam";}

    return 0;
}

```

4. Futtasd lépésenként az alábbi programot a tanártól kapott bemenetre! Értelmezd az eredményt!

```

#include <iostream>

using namespace std;

int main(){
    int i=0;
    double szam, osszeg = 0;

    while ( 1 ){
        cin >> szam;
        if ( !szam ) { break; }
        if ( szam < 0 ) { continue; }
        osszeg += szam;
        ++i;
    }

    if (i){ cout << "Atlag = " << osszeg / i;}
    else { cout << "Nem volt szam";}

    return 0;
}

```

5. Futtasd lépésenként az alábbi programot a tanártól kapott bemenetre! Értelmezd az eredményt!

```

#include <fstream>

```

```

#include <iostream>
#include <stdbool.h>

using namespace std;

int main(){
    ifstream fin; fin.open("szamsor.txt");
    if (!fin.is_open()) {
        cout << "Sikertelen input-file megnyitas!";
        return 0;
    }

    int x, n, i;

    fin >> n;

    bool b = false;
    for ( i = 1 ; i <= n ; ++i){
        fin >> x;
        if (!x) {b = true; break;}
    }

    if (b) {cout << "VAN"}
    else {cout << "NINCS"}

    fin.close();

    return 0;
}

```

6. Futtasd lépésenként az alábbi programot a tanártól kapott bemenetre! Értelmezd az eredményt!

```

#include <stdbool.h>
#include <iostream>

using namespace std;

int main(){
    int x, min, n, i;

    cin >> n;

    cin >> min;
    for ( i = 2 ; i <= n ; ++i){
        cin >> x;
        if (x < min) {min = x;}
    }

    cout << "MIN = " << min;

    return 0;
}

```

## HÁZI-feladatok:

1. Amire nem maradt idő laboron.
2. Nézz utána, hogyan lehet véletlen-számokat generálni, és old meg az alábbi feladatot: Beolvasunk a billentyűzetről egy  $n$  értéket, generálunk egy  $n$  elemű véletlen számsorozatot (melynek eleme a  $[0..99]$  intervallumba esnek), majd kiíratjuk a képernyőre külön a párosok és külön páratlanok átlagát.

## 3. feladatsor közép-haladóknak:

1. Generáljuk a négyzetszámokat  $n$ -ig: 1, 4, 9, 16, ...
2. Generáljuk az első  $n$  négyzetszámot az alábbi rekurzív képlet alapján:  $x_1 = 1$ ;  $x_n = x_{n-1} + (2n-1)$ , ha  $n > 1$ .
3. Generáljuk az alábbi szám-négyzet, illetve szám-háromszög sorozatok első  $n$  elemét:

1

22

22

333

333

333

...

1

22

12

333

223

123

...

1

12

22

123

223

333

...

1

2

22

3

33

333

...

1

12

22

223

233

333

...