

Programozás I.

Szekeres György

Eddig tanultuk

- Kiírás képernyőre
- Ékezetes karakterek kezelése
- Változók használata
- Matematikai műveletek
- Beolvasás konzolról
- Float, double, char
- If-else elágazás
- For ciklus

Relációs operátorok

Operátor	Operandus	Művelet
==	2	Két mennyiség egyenlőségét vizsgálja, nem összetévesztendő az értékadó operátorral!
!=	2	Két mennyiség különbözőségét vizsgálja, szokás "nem egyenlő" operátornak is nevezni
>	2	Nagyobb operátor, $a > b$ (a nagyobb-e mint b)
>=	2	Nagyobb egyenlő operátor, $a > b$ (a nagyobb-e vagy egyenlő mint b)
<	2	Kisebb operátor, $a < b$ (a kisebb-e mint b)
<=	2	Kisebb egyenlő operátor, $a < b$ (a kisebb-e vagy egyenlő mint b)

Logikai operátorok

Operátor	Operandus	Művelet
&&	2	Logikai „és”: kifejezés igaz lesz, ha mindkét kifejezés igaz (1 vagy I vagy true), minden más esetben, a művelet értéke hamis (0 vagy H vagy false) lesz
	2	Logikai "vagy": A kifejezés igaz lesz, ha valamelyik vagy mindkét kifejezés értéke igaz (1 vagy I vagy true), és a művelet értéke hamis (0 vagy H vagy false) lesz, ha mindkét kifejezés hamis
!	1	Logikai "nem": negáció, azaz ha az értéke 1, akkor 0-t ad vissza, ha pedig 0, akkor 1-et ad vissza

1. feladat

- Kérje be egy hallgató programozás jegyét, és írja ki szövegesen a képernyőre!

1. feladat

```
#include <iostream>

using namespace std;

int main() {
    setlocale(LC_ALL, "hun");
    cout<<"Adja meg a hallató programozás jegyét: ";
    int jegy;
    cin>>jegy;
    if(jegy==1) cout<<"Elégtelen";
    if(jegy==2) cout<<"Elégséges";
    if(jegy==3) cout<<"Közepes";
    if(jegy==4) cout<<"Jó";
    if(jegy==5) cout<<"Jeles";
    return 0;
}
```

2. feladat

- Kérje be egy hallgató programozás jegyét, és írja ki szövegesen a képernyőre!
- Használjon egymásba ágyazott if-else ágakat

2. feladat

```
#include <iostream>

using namespace std;

int main() {
    setlocale(LC_ALL, "hun");
    cout<<"Adja meg a hallató programozás jegyét: ";
    int jegy;
    cin>>jegy;
    if(jegy==1) cout<<"Elégtelen";
    else if(jegy==2) cout<<"Elégséges";
    else if(jegy==3) cout<<"Közepes";
    else if(jegy==4) cout<<"Jó";
    else if(jegy==5) cout<<"Jeles";
    return 0;
}
```


Switch-case

- Szintaxisa:
 - switch(kifejezés)
 - {
 - case 1: utasítás(ok)
 - case n: utasítás(ok)
 - default: utasítás(ok)
 - }
- A switch zárójelében lévő változó kiértékelése után a megfelelő case ág lefut
- break;

3. feladat

- Kérje begy egy hallgató programozás jegyét, és írja ki szövegesen a képernyőre!

3. feladat

```
#include <iostream>

using namespace std;

int main() {
    setlocale(LC_ALL, "hun");
    cout<<"Adja meg a hallató programozás jegyét: ";
    int jegy;
    cin>>jegy;
    switch(jegy) {
        case 1:
            cout << "Elégtelen";
            break;
        case 2:
            cout << "Elégséges";
            break;
        case 3:
            cout << "Közepes";
            break;
        case 4:
            cout << "Jó";
            break;
        case 5:
            cout << "Jeles";
            break;
        default:
            cout << "Hibás adat";
            break;
    }
    return 0;
}
```

While

- Szintaxisa:
 - while(kifejezés)
 - {
 - Ciklusmag;
 - }
- A ciklusmag mindaddig ismétlődik, amíg a vizsgált kifejezés értéke igaz
- A kifejezés kiértékelése a ciklusmag lefutása előtt történik meg
- Használata: Ha azt szeretnénk, hogy csak akkor hajtsódjon végre, ha a kifejezés igaz

4. feladat

- Kérjen be egy számot, és írja ki 1-től a bekért számig az összeget!

4. feladat

```
#include <iostream>

using namespace std;

int main() {
    setlocale(LC_ALL, "hun");
    cout<<"Adjon meg egy számot: ";
    int n, osszeg = 0;
    cin>>n;
    while (n>0)
    {
        osszeg += n;
        n--;
    }
    cout << "Az összeg: " << osszeg;
    return 0;
}
```

5. feladat

- Kérjen be egy számot, és döntse el, hogy a számnak hány valós osztója van!

5. feladat

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    setlocale(LC_ALL, "hun");
```

```
    cout << "Adjon meg egy számot: ";
```

```
    int n, i = 2, osztok = 0;
```

```
    cin >> n;
```

```
    while (i <= n / 2)
```

```
    {
```

```
        if(n % i == 0)
```

```
        {
```

```
            osztok++;
```

```
        }
```

```
        i++;
```

```
    }
```

```
    cout << "Osztók száma: " << osztok << endl;
```

```
    return 0;
```

```
}
```


Do-While

- Szintaxisa:
 - do
 - {
 - ciklusmag
 - } while(feltétel);
- A ciklusmagot képző utasítás(ok) végrehajtása után kerül sor a kifejezés (feltétel) vizsgálatára!
- Használata: ha azt szeretnénk, hogy a ciklusmag egyszer mindenképpen lefusson
- Mindig az utasítás végrehajtása után következik a kifejezés kiértékelése , amely ha igaz, akkor új iteráció kezdődik, ha hamis, akkor a ciklus befejezi futását

6. feladat

- Kérje be a pontszámot 0-100 között. Ha 50 pont alatt van, írja ki, hogy megbukott, ellenkező esetben "Aláírva!" jelenjen meg!

6. feladat

```
#include <iostream>

using namespace std;

int main() {
    setlocale(LC_ALL, "hun");
    int pont;
    do
    {
        cout << "Adjon meg egy pontszámot 0-100 között: ";
        cin >> pont;
    } while (pont < 0 || pont > 100);
    if(pont<50)
        cout<<"Megbukott";
    else
        cout<<"Aláírva!";
    return 0;
}
```

Tömbök

- A tömb (array) típus olyan adatok halmaza, amelyek azonos típusúak, és a memóriában folytonosan helyezkednek el
- Előre meghatározott számú eleme van
- Valójában egy változó készlet
- Az adatokat a változó neve után írt elemszámmal, azaz az index segítségével érjük el
- A leggyakrabban használt tömbtípus az egyetlen kiterjedéssel (dimenzióval) rendelkező egydimenziós tömb. Vektornak is szokás nevezni.
- Szintaxisa:
 - típus tömbnév [méret];

Tömbök definiálása, deklarálása

- Definiálás:
 - `int tmb[8];`
 - Ez egy 8 elemű egész típusú tömb.
- Deklarálás:
 - `int tmb[8] = { 1, 2, 3, 4, 2, 1, 4, 5 };`
 - `char tmb[7] = { 'a', 'b' };`
 - Az 1-2. elemnek adtunk csak értéket, a többi értéke 0 vagy auto
 - `float tmb[] = { 12.3, 15.4, 36.7 };`
 - A tömb mérete a megadott értékek számával egyenlő lesz, jelen esetben 3
- Indexelés:
 - A tömböket 0-tól indexeljük
 - `int tmb[4];`
 - `tmb[0] = 56;`
 - A tömb első, azaz 0. eleme 56 lesz

7. feladat

- Készítsen egy tömböt, és írassa ki az elemeit!

7. feladat

```
#include <iostream>

using namespace std;

int main() {
    setlocale(LC_ALL, "hun");
    int tmb[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    for (int i = 0; i < 10; i++)
    {
        cout << "A tömb " << i << ". eleme: " << tmb[i] << endl;
    }
    return 0;
}
```

8. feladat

- Írassa ki az 5-tel vagy a 7-tel osztható számokat!
- Írassa ki az 5-tel és a 7-tel osztható számokat!

8. feladat

```
#include <iostream>

using namespace std;

int main() {
    setlocale(LC_ALL, "hun");

    int n;

    cout << "Adjon meg egy számot: ";
    cin >> n;

    cout << "1-től " << n << "-ig a következő számok oszthatóak 5-tel vagy 7-tel: "
        << endl;

    for (int i = 1; i <= n; i++)
    {
        if (i % 5 == 0 || i % 7 == 0)
        {
            cout << i << " ";
        }
    }
}
```

```
cout << "1-től " << n << "-ig a következő számok oszthatóak 5-tel és 7-tel: " <<
    endl;

for (int i = 1; i <= n; i++)
{
    if (i % 5 == 0 && i % 7 == 0)
    {
        cout << i << " ";
    }
}

return 0;
}
```