

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
- While, do-while ciklusok
- Switch-case elágazás
- Tömbök

Karakterkezelő függvények

- toupper() - nagybetűssé alakítja a karaktert
- tolower() - kisbetűssé alakítja a karaktert
- isalpha() - megvizsgálja, hogy a karakter betű-e
- isdigit() - megvizsgálja, hogy a karakter szám-e

1. feladat

- Írjunk egy programot, amely bekér egy betűt, és ennek írassuk ki a kis illetve nagybetűs változatát is

1. feladat

```
#include <iostream>
using namespace std;

int main() {
    setlocale(LC_ALL, "hun");
    char betu;
    cout<<"Adjon meg egy betűt: ";
    cin>>betu;
    cout<<"A megadott betű: "<<tolower(betu)<<", nagybetűs alakja: "<<toupper(betu)<<endl;
    return 0;
}
```

2. feladat

- Írjon egy programot, amely eldönti egy karakterről, hogy az szám, betű vagy egyéb karakter

2. feladat

```
#include <iostream>
using namespace std;

int main() {
    setlocale(LC_ALL, "hun");
    char betu;
    bool betu, szam;
    cout<<"Adjon meg egy karaktert: ";
    cin>>betu;
    betu = isalpha(betu);
    szam = isdigit(betu);
```

```
    if(betu)
        cout<<"A megadott karakter egy betű."<<endl;
    else if(szam)
        cout<<"A megadott karakter egy szám."<<endl;
    else
        cout<<"A megadott karakter egyéb
karakter."<<endl;
    return 0;
}
```

Többdimenziós tömbök

- Szokás ezeket a tömbök tömbjeinek is nevezni (mátrixok)
 - Vannak sorai és oszlopai → táblázatnak is szokás nevezni
- Szintaktika:
 - Típus név [sorok száma] [oszlopok száma];
 - Pl.: `int tmb[5][3]`
- Értékadás:
 - `tmb[2][0] = 6;`
 - Ez az 1. oszlop 3. sorának értékét állítja be 6-ra

Karaktertömb

- Karakter típusú egydimenziós tömbök

```
#include <iostream>
using namespace std;
```

```
int main() {
    setlocale(LC_ALL, "hun");
    char nev[25];
    cout<<"Adja meg a nevét: ";
    cin>>nev;
    cout<<"Üdvözöllek "<<nev<<"!"<<endl;
    return 0;
}
```

- Csak szóközig olvas a cin, tehát nem lesz jó megoldás a nevekhez

cin.getline()

- Karakterek kiolvasása karaktertömbbe megadott hosszon

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
    setlocale(LC_ALL, "hun");  
    char nev[25];  
    cout<<"Adja meg a nevét: ";  
    cin.getline(nev, 25);  
    cout<<"Üdvözöllek "<<nev<<"!"<<endl;  
    return 0;  
}
```

- Csak szóközig olvas a cin, tehát nem lesz jó megoldás a nevekhez

Karakterkezelő függvények

- Használatukhoz szükséges a *cstring* header állomány
- `strcpy_s(tmb1, tmb2);` - tmb2 tömb átmásolása a tmb1-be
- `strcmp(tmb1, tmb2);` - tömbök összehasonlítása, ha tmb1 megegyezik tmb2-vel, akkor 0-t ad vissza (kis- és nagybetű nem számít)
- `strlen(tmb);` - karaktertömb hosszát adja vissza (szóköz is beleszámít)
- `cin.clear();` - billentyűzet puffer ürítése, összes változót alaphelyzetbe állítja
- `cin.ignore();` - figyelmen kívül hagyja az olyan olvasatlan karaktereket, amelyek a memória pufferben maradtak (pl.: `\n`-t, csak egy karaktert)

3. feladat

- Kérjen be egy nevet, amit kiírat nagybetűsen, kisbetűsen és hasonlítsa össze őket

3. feladat

```
#include <iostream>
#include <cstring>
using namespace std;

int main() {
    setlocale(LC_ALL, "hun");
    char nev[25], nev2[25];
    cout << "Adja meg a nevét: ";
    cin.getline(nev, 25);
    strcpy(nev2, nev);
```

```
    for (int i = 0; nev2[i] != '\0'; i++) {
        nev2[i] = toupper(nev2[i]);
    }
    cout << "Név nagybetűsen: " << nev << endl;
    for (int i = 0; i < strlen(nev2); i++) {
        nev2[i] = tolower(nev2[i]);
    }
    cout << "Név kisbetűsen: " << nev << endl;
    bool egyforma = strcmp(nev2, nev) == 0;
    cout << "Az eredeti és a kisbetűs név " << (egyforma ?
        "meg egyezik" : "nem egyezik meg") << "." << endl;
    return 0;
}
```

4. feladat

- Kérjen be egy 2 soros verset, soronként max 60 karakter, majd írja ki nagybetűsen és számolja meg a szóközöket

4. feladat

```
#include <iostream>
#include <cstring>
using namespace std;

int main() {
    setlocale(LC_ALL, "hun");
    char vers[2][60], bet;
    int i, j, szokoz = 0;
    for (i = 0; i < 2; i++)
    {
        cout<< "Adja meg a vers " << i + 1 << ". sorat: " << endl;
        cin.getline(vers[i], 60);
    }
    cout << endl;
```

```
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < strlen(vers[i]); j++)
        {
            if (vers[i][j] == ' ')
                szokoz++;
            bet = toupper(vers[i][j]);
            cout << bet;
        }
        cout << endl;
    }
    cout << endl << "A szokozok szama: " << szokoz << endl;
    return 0;
}
```

Struktúra típus

- Szintaxisa:
 - `struct nev`
 - `{`
 - `típus név[méret ha kell];`
 - `};`
- Olyan típus, amely több, tetszőleges típusú adatok együttese
- A struktúra elemek (adattagok) saját névvel rendelkeznek
- `struct` kulcsszó a `main`-en kívül

Struktúra típus

- Struktúratípus deklaráció után változókat is létrehozhatunk
 - Pl.: adat gyerek, felnőtt;
- Az adattagok a deklaráció sorrendjében tárolódnak
- Hivatkozás struktúra elemre:
 - azonosító.adattag;
 - Pl.: gyerek.magassag;
- Értékadás struktúra elemnek:
 - azonosító.adattag = érték;
 - Pl.: gyerek.suly = 54.6;

5. feladat

- Írja ki a megadott struktúra típusú változó adatait (NE tömbbel)
- struct test
- {
- char nev[25];
- int magassag;
- float suly;
- };

5. feladat

```
#include<iostream>
using namespace std;
```

```
struct test
{
    char nev[25];
    int magassag;
    float suly;
};
```

```
int main()
{
    test gyerek;
    cout << "Add meg a gyerek nevet: " << endl;
    cin.getline(gyerek.nev,25);
    cout << "Add meg a gyerek magassagat cm-ben: "<<endl;
    cin >> gyerek.magassag;
    cout << "Add meg a gyerek sulyat kg-ban: " << endl;
    cin >> gyerek.suly;
    cout<<"A gyerek adatai a kovetkezők:"<<endl;
    cout<<gyerek.nev<<" - "<<gyerek.magassag<<" cm es "<<
gyerek.suly <<" kg"<<endl;
    return 0;
}
```

6. feladat

- Írja ki a megadott struktúra típusú változó adatait tömbbel
- struct test
- {
- char nev[25];
- int magassag;
- float suly;
- };

6. feladat

```
#include<iostream>
using namespace std;
```

```
struct test
{
    char nev[25];
    int magassag;
    float suly;
};
```

```
int main()
{
    test gyerek[3];
```

```
    for (int i = 0; i < 3; i++)
    {
        cout << "Adja meg a(z) " << i + 1 << ". gyerek nevét: ";
        cin.getline(gyerek[i].nev, 25);

        cout << "Adja meg a(z) " << i + 1 << ". gyerek magasságát: ";
        cin >> gyerek[i].magassag;

        cout << "Adja meg a(z) " << i + 1 << ". gyerek súlyát: ";
        cin >> gyerek[i].suly;

        cin.ignore();
    }

    for (int i = 0; i < 3; i++)
    {
        cout << i + 1 << ". gyerek adatai:" << endl;

        cout << "Név: " << gyerek[i].nev << endl;

        cout << "Magasság: " << gyerek[i].magassag << " cm" << endl;

        cout << "Súly: " << gyerek[i].suly << " kg" << endl;
    }

    return 0;
}
```