

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
- Többdimenziós tömbök
- Struktúrák

String típus

- karakterlánc típus (valójában egy osztály)
- használatához szükség van a *string* header állományára
- definiálás: string azonosító; pl.: `string str;`
- deklarálás: string azonosító = "szöveg"; pl.: `string str = "valami";`
- Karakterek számozása 0-tól kezdődik
- beolvasás: `getline(cin, azonos);` pl.: `getline(cin, str);`
- műveletvégzéshez a következő formula kell: `stringváltozó.művelet(argumentumok);`

String osztály tagfüggvények

- `név.length();` - string hossza
- `név.at(i);` - string *i*. karakteréhez való hozzáférés
- `név.clear();` - string kiürítése
- `név.erase(honnan, mennyit);` - stringrész törlése
- `név.substr(honnan, mennyit);` - stringrész kivágása
- `név.append("szöveg");` - szöveg hozzáfűzése stringhez
- `név.insert(honnan, "szöveg");` - szöveg beillesztése adott pozíciótól
- `név.replace(honnan, mennyit, "szöveg");` - stringrész lecserélése
- `név.find("szöveg", honnan);` - szöveg keresése adott pozíciótól, ha nem adjuk meg elejétől keres, az első találatig
- `név.rfind("szöveg", honnan);` - szöveg utolsó előfordulását keresi a honnantól visszafel
- `név.compare("szöveg");` - adott szöveg összehasonlítása stringgel, megegyezik 0
- `név.empty();` - üres-e a string, ha üres (igaz) 1

1. feladat

- Kérjen be egy nevet, aminek kiírja a hosszát, átmásolja egy stringbe, kiíratás után megkérdezi akarja-e törölni a másolatot és összehasonlítja!

1. feladat

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

int main()
{
    char b;
    string nev1, nev2;
    cout << "Adja meg a nevet ekezetek nélkül: ";
    getline(cin, nev1);
    nev2 = nev1; // a nev1 átmásolása nev2-be
    cout << "Az Ön nevének a masolata: " << nev2 << endl;
```

```
    cout << "A neve hossza a szókozzal együtt: " <<
    nev1.length() << endl;
    cout << "Szeretne torolni a masolatot? igen= i, nem = n: ";
    cin >> b;
    if(b=='i') nev2.clear();
    if (nev1==nev2) cout<<"A ket karakterlanc
    egyforma."<<endl;
    else cout << "A ket karakterlanc mar nem egyforma."<<
    endl;
    return 0;
}
```


2. feladat

- Kérjen be egy közmondást. Írattassa ki hogy a közmondás szóközzel együtt hány karakterből áll, fűzze hozzá - közmondás; másolathoz -ez is; Keresse meg a 'kapni' szót és írja ki hogy hányadik karaktertől kezdődik, majd az utolsó x karaktert, amit kér kiírja.

2. feladat

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

int main()
{
    int x,y;
    string str, potstr;
    cout << "Add meg a kedvenc kozmondasod: " << endl;
    getline(cin, str);
    potstr = str;
    cout <<"Ez a mondas szokozokkal "<<str.length()<<" karakterbol  
all."<<endl;
    cout << str.append(" - kozmondas") << endl;
```

```
    potstr += " - ez is"; //hozzafuzes operátorral
    cout << potstr<<endl;
    cout << str.find("kapni")+1 << ". karaktertol kezdodik a 'kapni' szo."
    << endl;
    cout << "Hanyadik karakterre vagy kivancsi? " << endl;
    cin >> x;
    cout <<"A kozmondas "<<x<<". karaktere: "<<str.at(x-1)<<endl;
    cout <<"A mondas hany utolso karakteret szeretned latni? "<<
        endl;
    cin >> y;
    cout<<"A kozmondas utolso "<<y<<" karaktere: "<<
    str.substr(str.length()-y,y)<<endl;
    return 0;
}
```


3. feladat

- Bekéri 3 könyv szerzőjét, címét és árát egy struktúra tömbbe, majd véletlenszámgenerátorral kisorsolja, melyik a nyertes és kiírja.

3. feladat

```
#include <iostream>
#include <string>
using namespace std;
struct konyv
{
    string szerzo;
    string cim;
    int ar;
};
int main()
{
    konyv gyerek[3];
    int j=rand() % 3 + 1; //értékadás véletlenszámgenerátorral 1-3
    között
```

```
for (int i = 0; i < 3; i++) //adatok bekérése
{
    cout << "Adja meg a(z) " << i + 1 << ". gyerek konyv szerzojet: ";
    getline(cin, gyerek[i].szerzo);
    cout << "Adja meg a konyv cimet: ";
    getline(cin, gyerek[i].cim);
    cout << "Adja meg a konyv arat: ";
    cin >> gyerek[i].ar;
    cin.ignore();
}
system("cls"); //képernyőtörlés
cout << "A sorsolas eredménye: " << j << endl;
cout << "A nyertes konyv szerzoje: " << gyerek[j].szerzo << endl;
cout << "A nyertes konyv cime: " << gyerek[j].cim << endl;
cout << "A nyertes konyv ara: " << gyerek[j].ar << endl;
return 0;
}
```

4. feladat

- Egy hallgatói csoportba max. 15-en járnak. Tároljuk el a hallgatók nevét, jegyét és a születési évét!
- Kérjük be a valós csoportlétszámot ellenőrzötten majd a hallgatói adatokat.
- Számítsuk ki a csoport átlagát, majd írassuk ki a legöregebb hallgató nevét!

4. feladat

```
#include<iostream>
#include<string>
using namespace std;
struct tanulo
{
    string nev;
    int jegy;
    int szul;
};
```

```
int main()
{
    tanulo hallg[15];
    int letszam, i, atlag = 0;
    do // letszam bekérése ellenőrzötten
    {
        cout << "adja meg a csoport letszamot: ";
        cin >> letszam;
        cin.ignore();
    } while (letszam < 1 || letszam>15);
    // folytatás
```

4. feladat

```
for (i = 0; i < letszam; i++)// hallgatói adatok ellenőrzöten
{
    cout << "adja meg a(z) " << i + 1 << ". tanulo nevet: ";
    getline(cin, hallg[i].nev);
    cout << "adja meg a születési evet: ";
    cin >> hallg[i].szul;
    do
    {
        cout << "adja meg a jegyet: ";
        cin >> hallg[i].jegy;
        cin.ignore();
    } while (hallg[i].jegy < 1 || hallg[i].jegy > 5);
}
```

```
system("cls"); //képernyő törlés

for (i = 0; i < letszam; i++)
{
    atlag += hallg[i].jegy;
}

cout << "\nA csoport atlaga: " << (float)(atlag / letszam) << endl;
return 0;
}
```