Uvod u programiranje - stringovi

1. Šta smo naučili na prošlom času?
http://bee.bubblecup.org/Problems/NZS#problem
http://bee.bubblecup.org/Problems/-22

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
string s;
    cin >> s;
    cout << s << endl;

    return 0;
}
```

Uvežbavamo stringove

http://bee.bubblecup.org/Lekcije/stringovi-1
http://bee.bubblecup.org/Lekcije/stringovi-1#problemi

1. Napisati C++ program koji će ispisati ime i prezime Miki Maus u dva reda, dok u 3. redu će ispisati ime i prezime sa tri uzvičnika na kraju!!!
Ulaz: **nema.**
Izlaz: **tri reda na standardnom izlazu.**
Primer

Ulaz	Izlaz:
	Miki Maus
	Miki Maus
	Miki Maus!!!

```
#include <iostream>
#include <string>

using namespace std;

int main(){
  string ime="Miki";
  string prezime("Maus");
  cout << ime << " " << prezime << endl;
  cout << ime +  " " + prezime << endl;
  string spojeno=ime + " " + prezime;
  spojeno=spojeno+'!';
  spojeno+="!!";
  cout << spojeno << endl;
    return 0;
}
```

U C++-u postoji tip podataka string, koji sadrži tekst. On se koristi kao i svi ostali tipovi, te se neka promenljiva tipa string deklariše na sledeći način:

```
using namespace std;
string s = "Bubble bee string";
```

Ili samo

```
std::string s = "Bubble bee string";
```

Možete da primetite da se navodnici koriste za dodelu teksta kao vrednosti stringu u primeru iznad. String ovde predstavlja nisku (niz) pojedinačnih znakova.

Dodela stringa

Takođe, možemo da dodelimo vrednost jednog stringa drugom:

```
string str1 = "Zdravo!";
string str2;
str2 = str1;
```

Možete i da inicijalizujete jedan string drugim kao u primeru:

```
string str1 = "Zdravo!";
string str2 = str1;
```

"Spajanje" dva stringa

Dva stringa (ili više njih) se mogu spojiti koristeći operaciju sabiranja "+" kao što smo radili sabirajući brojeve.

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string s = "Bubble";
    string p = "bee";
    s = s + p;
    cout << s << endl;
    return 0;
}
```

Nakon izvršavanja programa promenljiva `s` će sadržati string "`Bubblebee`".

Unos stringa sa tastature

Kao i ostali tipovi u C++-u, za učitivanje i ispisivanje stringa se koriste se globalni objekti `cin` i `cout`.

Primer unošenja stringa sa tastature:

```
std::string moj_string;
cin>>moj_string;
```

Operator >> će zaustaviti naš program i sačekati da korisnik nešto otkuca. Evo jednog primera unosa:

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

int main(){
  string ime;
  cout << "Unesite svoje ime: ";
  cin >> ime;
  cout << "Zdravo, " << ime << "!" << std::endl;
```

```
  return 0;
}
```

Objekat cin međutim prestaje da učitava string čim naiđe na razmak, tab (tabulator) ili nov red. Ukoliko želiš da uneseš celu liniju teksta možeš da koristiš funkciju *getline*:

```
getline(cin, moj_string);
```

Prvi argument funkcije getline je cin, koji kaže odakle nam dolazi tekst (u ovom slučaju sa konzole, to jest tastature). Drugi argument je ime string promenljive u koju želiš da smestiš uneti tekst. `getline` čita celu liniju sve dok korisnik ne pritisne Enter. Ovo je korisno kada ulazni string sadrži razmake. Sad zamisli da od korisnika tražiš da unese puno ime i prezime u jednoj liniji, pošto ćemo tako imati razmak u tekstu koristićemo `getline` .

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

int main(){
  string puno_ime;
  cout << "Unesite svoje puno ime i prezime: ";
  getline(cin, puno_ime);
    cout << "Zdravo " << puno_ime << "!" << std::endl;

  return 0;
}
```

Evo za nekoliko primera ulaznog teksta koje bi rezultate uporedo dali cin i getline:

"ana bane"
cin: "ana"
getline: "ana bane"

"ana bane
cane dejan"
cin: "ana"
getline: "ana bane"

"ana"
cin: "ana"
getline: "ana"

Dužina stringa

Slova, razmaci, znaci interpunkcije i svi ostali znaci koji se koriste u računaru nazivaju se karakteri. Da bismo dobili broj karaktera u nekom stringu, pišemo:

```
str1.length()
```

Zapis str1.length() se koristi kao (brojni) izraz, čija je vrednost upravo broj karaktera stringa str1, odnosno njegova dužina.

Evo jednog primera kratkog programa u kome ilustrujemo dodelu, spajanje i dužinu stringa:

```
#include <iostream>
#include <string>

using namespace std;

int main()

{
 string ime;
 string prezime;
 string puno_ime;
 int duzina_imena;

 cout << "Unesi svoje ime: ";
 getline(cin, ime);

 cout << endl;

 cout << "Unesi svoje prezime: ";
 getline(cin, prezime);
 puno_ime = ime + " " + prezime;
 duzina_imena = puno_ime.length() - 1; // Oduzimamo 1 zbog razmaka

 cout << endl <<endl;
 cout << "Zdravo, " << puno_ime << endl;
 cout << "Tvoje puno ime sadrzi " << duzina_imena << " karaktera." << endl;

    return 0;

}
```

Primer ispisa ovog programa:

Unesi svoje ime: Ibro
Unesi svoje prezime: Dirka
Zdravo, Ibro Dirka
Tvoje puno ime sadrzi 9 karaktera.

Poređenje stringova

Sledeći programski kod određuje da li je ulazni string jednak unapred zadatom stringu (stringu naše lozinke):

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

int main()
{
    string lozinka;
    getline(cin, lozinka);

    if(lozinka == "3astoL1$am0dabrao0vol1koDugackuLoz1nku")
    {
      cout<<"Tacna lozinka!";
    }
    else
```

```
    {
      cout<<"Netacna lozinka!";
    }

    return 0;
}
```

Dakle, operatori poređenja `>, <, ==, >=, <=, !=` se mogu primeniti i na stringove tako da možemo formirati izraze koji vraćaju vrednosttrue/false. Operatori <, >,<= i >=, upoređuju stringove leksikografski, tj. po abecedi, a ne po dužini stringa. Tako bi npr. string "akrobacije" bio "manji" od stringa "burek", iako je duži od drugog stringa, zato što se pre njega pojavljuje u rečniku.

Toliko za sad o stringovima.

ZADACI ZA VEŽBU

2. Šta je rezultat rada programa? Zašto?

```
#include <iostream>
#include <string>

using namespace std;

int main(){
  string niz1, niz2;  //prazni su
  string niz3="111";
  string niz4(3,'8'); // niz "888"

   niz1="123";
   niz2 += niz3+" 333 "+"_"+niz1;
   niz4 += niz2;
   cout << niz2+"!"+'\n'+niz4<< endl;
  if (niz1=="123" && niz4=="888"+niz2) cout << "jednake vrednosti" << endl;
  else cout << "nejednake vrednosti" << endl;
    return 0;
}
```

Izlaz
111 333 _123!
888111 333 _123
jednake vrednosti

3. // string::length

```
#include <iostream>
#include <string>

int main ()
{
  std::string str ("Mika Pera Laza");
  std::cout << "Velicina stringa iznosi " << str.length() << " karaktera.\n";
  return 0;
}
```

IZLAZ
Velicina stringa iznosi 14 karaktera.

4. **// string::size**

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

```
int main ()
{
  string str ("Pera Mika Laza");
  cout << "Velicina stringa isnosi " << str.size() << " bajtova-karaktera.\n";
  return 0;
}
```

IZLAZ
Velicina stringa isnosi 14 bajtova-karaktera.

5.
```
// string::find
  // trazenje po stringu

#include <iostream>       // std::cout
#include <string>         // std::string
```
using namespace std;

```
int main ()
{
  string str ("Milovana voli Ana Ana.");
  string str2 ("Ana");

  // razlicite verzije pretrage find, rfind, find_first_of, find_last_of
  cout << str.find(str2) << endl;
  cout << str.rfind(str2) << endl;
  cout << str.find_first_of("aeiou") << endl;
  cout << str.find('l') << endl;
  cout << str.find_last_of("aeiou") << endl;
cout << str.find_first_not_of("abc") << endl;
cout << str.find_last_not_of('l') << endl;

  return 0;
}
```

IZLAZ	OBJASNJENJE
14	//prva pojava Ane je od pozicije 14! Oprez, znak M je na poziciji 0
18	//druga pojava Ane je od pozicije 18
1	// prvi se pojavljuje znak a na poziciji 1
2	// znak l se pojavljuje na poziciji 2
20	// poslednja pojava znaka a je na poziciji 20
0	// znak M se razlikuje i od a, i od b i od c
21	//znak . je poslednji karakter (prvi otpozadi) koji se razlikuje od slova l

6. U stringu ab+cd=abcd nadji prvi znak koji ne pripada engleskoj abecedi.

```
// string::find_first_not_of
#include <iostream>       // std::cout
#include <string>         // std::string
```
using namespace std;

```
int main ()
{
```

```
  string str ("ab+cd=abcd");

  int nasao= str.find_first_not_of("abcdefghijklmnopqrstuvwxyz");

  if (nasao>=0)
  {
    cout << "Prvi znak koji ne pripada engleskoj abecedi je " << str[nasao];
    std::cout << " na poziciji " << nasao << '\n';
  }

  return 0;
}
```

IZLAZ
Prvi znak koji ne pripada engleskoj abecedi je + na poziciji 2

7. Napisati C++ program koji ce od stringa koji sadrzi sva velika slova engleske abecede napraviti string koji ne sadrzi prvih 5 slova.

```
#include <iostream>       // std::cout
#include <string>         // std::string
using namespace std;

int main ()
{
 string str ("ABCDEFGHIJKLMNOPQRSTUVWXYZ");

 cout << str << endl;
 str.erase (0,5);  //izbrisi karakter pocev od pozicije 0 i ostalih 5 karaktera
 cout << str << endl;

 return 0;
}
```

IZLAZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
FGHIJKLMNOPQRSTUVWXYZ

8. Napisati C++ program koji ce ukloniti beline sa kraja stringa „12345678 “.
SAVET: Koristite funkciju find_last_not_of i erase

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

int main ()
{
 std::string str ("12345678   \t");
 std::string beline (" \t\f\v\n\r");

 cout << '[' << str << "]\n";
 int nasao = str.find_last_not_of(beline);
 if (nasao >=0)
   str.erase(nasao+1);

 cout << '[' << str << "]\n";

 return 0;
```

}

IZLAZ
[12345678]
[12345678]

9. Prevodjenje malih slova stringa u velika slova

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
     string str1,str2;
     str1="!abcdefghijkl!";

//kopiranje stringova moze se obaviti i preklopljenim operatorom dodele vrednosti!
     str2=str1;

     cout<<"ovo je izvorni string: "<<str1<<endl;

     for (int i=0; str1 [i]; i++) str2 [i] = toupper (str1[i]);
     cout << str2 << endl;
     return 0;
}

IZLAZ
 ovo je izvorni string: !abcdefghijkl!
!ABCDEFGHIJKL!
```

10. Napisati program koji ucitava string i proverava da li je palindrom.

ULAZ IZLAZ
anavolimilovana Da
neven Da
oko Da
Ana Ne
12321 Da

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string s,t;
    int i;
    cin >>s;
    t=""; // postavljanje pocetne vrednosti stringa na "0", odnosno na prazan string


  //s.length() vraca duzinu stringa s, samim tim pocetna vrednost brojaca i je duzina
// stringa -1 (zato sto indeksi znakova u stringu pocinju od 0)

    for (i=s.length()-1;i>=0;i--)
    {
       t=t+s[i];// pravimo string koji se dobija citanjem stringa s unazad
    }

    if (s==t) cout <<"Da!";
    else cout <<"Ne!";
    return 0;
```

}

Napisite novo resenje koje ne koristi dodatni string t!!!