Ниска знакова

У програмском језику C++ постоје знаковни подаци и они се обележавају са char.
Овај тип података се користи за приказ појединачног знака.
За рад са већим бројем знакова у низу (текст) користи се ниска знакова (C-стринг) или се користи класа string.

Дефиниција ниске знакова

На крају сваке ниске знакова се налази празан знак који се аутоматски додаје иза последњег корисног знака у тексту.
Празан знак има вредност 0.
Овај знак се не може учитавати или исписивати и служи да би означио крај текста.
Обично се обележава са '\0'.
Често је неопходно унапред предвидети дужину ниске знакова да би се предвидело место за празан знак на крају текста.
Преводилац за C++ генерише по један цео број за сваки податак типа char и на крају додаје празан знак.
Празан знак има целобројну вредност 0.
Тако да је текст "zdravo" је исто што и низ појединачних знакова 'z' 'd' 'r' 'a' 'v' 'o' '\0'.

Иницијализација ниске знакова

char ime[dužina] = tekst
ime – назив ниске знакова,
dužina – цео број којим се одређује број знакова који може ниска знакова да садржи,
tekst – низ знакова под наводницима
Пример:
char niz[30] = "A sada, sve ponovo!";
char niz [] = "A sada, sve ponovo!";

Ако се дефинише дужина ниске знакова већа него што је број знакова у иницијализованој ниски знакова, преостали број елемената се допуњава са празним знацима тј. са нулама.
Ако се иницијализује ниска знакова без навођења димензије, тада се за дужину ниске знакова узима наведени низ и аутоматски додаје празан знак на крају ниске знакова.
Дужина ниске знакова у примеру је 20 (19 знакова у низу и 1 празан знак).
Сви знаци белине (бланко знак, ентер, таб) се такође сматрају знаком.
Већ једном дефинисана и иницијализована ниска знакова се не може поново дефинисати и иницијализовати.

Пример01: Написати програм који за унето име и презиме приказује оба податка у истом реду.

```
#include <iostream>
using namespace std;
int main ()
{
	char ime[15];
	char prezime[20];
	cout << "Unesi ime: ";
	cin >> ime;
	cout << "Unesi prezime: ";
	cin >> prezime;
	cout << ime << " " << prezime << endl;
	return 0;
}
```

```
Unesi ime: Petar
Unesi prezime: Petrovic
Petar Petrovic
```

Пример02: Написати програм који у стринг уписује сва велика слова енглеског језика и онда их исписује.

```
#include <iostream>
using namespace std;
int main ()
{
	char a[26];
	int i;
	for(i = 0; i < 26; i++) a[i] = 'A' + i;
	cout << "Slova engleske abecede:" << endl ;
	for(i = 0; i < 26; i++)  cout << a[i];
	cout<<endl;
	return 0;
}
```

Slova engleske abecede:
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Функције за обраду знаковних података

Функције за обраду знаковних података се могу поделити у три групе: одређивање врсте појединачних знакова, обраду ниске знакова, конвертовање ниске знакова у нумеричке типове података.

Функције за одређивање врсте знакова

За одређивање врсте знакова користи се библиотека cstring.

функција	**опис рада**
isalnum(promenjiva)	Да ли је промењива слово или цифра
isalpha(promenjiva)	Да ли је промењива слово
islower(promenjiva)	Да ли је промењива мало слово
isupper(promenjiva)	Да ли је промењива велико слово
isdigit(promenjiva)	Да ли је промењива децимална цифра

Функције за конверзију ниске знакова у нумеричке податке

Конверзија ниске знакова у нумеричке податке захтева библиотеку cstdlib.

функција	**опис рада**
atoi (string)	Претвара ASCII стринг у цео број. Ако стринг садржи децимални број функција враћа еквивалентан цео број. Испред децималног броја може бити произвољан број нула или празнина или табулација, може бити и негативан број, али ће функција да врати цео позитиван број.
atof (string)	Конверзија стринга у реалан број двоструке тачности.
atol (string)	Конверзија стринга у податак типа long.

Пример01: Написати програм који ће у ниски знакова заменити један знак.

```
#include <iostream>
using namespace std;
int main ()
{
	char a[] = "proba";
	cout << "originalni string: " << a << endl;
	a[3] = 'j';
	cout << "izmenjeni string: " << a << endl;
	return 0;
}
```

originalni string: proba
izmenjeni string: proja

Пример02: Написати програм који показује рад са функцијама за обраду знаковних података

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
	char a[] = "c3PAo..?";
	int i = 0;
	while (isalnum(a[i])) i++;
	cout << "Prva " << i << " znaka su alfanumerici." << endl;
	return 0;
}
```

```
Prva 5 znaka su alfanumerici.
Press any key to continue . . .
```

Пример03: Написати програм који показује рад са функцијама за обраду знаковних података

```
#include<iostream>
#include<cstdlib>
using namespace std;
int main()
{
	char niz[100];
	cout << "Unesi ceo broj: ";
	cin >> niz;
	int broj = atoi(niz);
	cout << "Broj je " << broj << endl;
	int suma = atoi(niz) + 10;
	cout << "Zbir broja i 10 je " << suma << endl;
	return 0;
}
```

```
Unesi ceo broj: 3
Broj je 3
Zbir broja i 10 je 13
Press any key to continue . . .
```

Функције за обраду ниске знакова

За обраду ниске знакова потребно је укључити библиотеку cstring. Код употребе ових функција мора се обезбедити довољно места у резултујућој промењивој како би прихватила почетне стрингове и празан знак.

функција	**опис рада**
strcpy(niz1, niz2)	Преписује ниску2 у ниску1
strcat(niz1, niz2)	Иза ниске1 додаје се ниска2
strlen(niz)	Одређује дужину ниске знакова без празног знака
strcmp(niz1, niz2)	Резултат ове функције је цео број. Ако су две ниске знакова исте по абецедном редоследу вредност ове функције је 0; ако је ниска1 испред ниске2 онда је вредност мања од 0; ако је ниска2 испред ниске1 онда је вредност већа од 0.
strchr(niz, znak)	Вредност ове функције је показивач на први елемент ниске који садржи знак. Ако тај знак не постоји у нисци, онда је вредност NULL.
strrchr(niz, znak)	Вредност ове функције је показивач на последњи елемент ниске који садржи знак. Ако тај знак не постоји у нисци, онда је вредност NULL.
strstr(niz, znak)	Вредност ове функције је показивач на први елемент ниске где се знак појављује као подниз. Ако тај знак не постоји у нисци, онда је вредност NULL.

Пример01: Одредити број знакова у нисци знакова унетог са тастатуре.

```
#include <iostream>
#include <cstring>
using namespace std;
int main ()
{
	char a[100];
	cout << "Unesi rec: ";
	cin >> a;
	cout << "Uneta rec ima duzinu " << strlen(a) << endl;
	return 0;
}
```

```
Unesi rec: komplikacija
Uneta rec ima duzinu 12
```

Пример02: У унетој нисци знакова претворити мала слова у велика.

```
#include <iostream>
#include <cstring>
using namespace std;
int main ()
{
	char a[100];
	cout << "Unesi rec: ";
	cin >> a;
	for(int i = 0; i < strlen(niska); i++) a[i] = toupper(a[i]);
	cout << "Uneta rec sa velikim slovima: " << a << endl;
	return 0;
}
```

```
Unesi rec: KoMpLiKaCiJa
Uneta rec sa velikim slovima: KOMPLIKACIJA
```

Пример03: У унетој нисци знакова претворити слово а у @.

```
#include <iostream>
#include <cstring>
using namespace std;
int main ()
{
	char a[100];
	cout << "Unesi rec: ";
	cin >> a;
	for(int i = 0; i < strlen(a); i++)
	{
		if (a[i] == 'a') a[i] = '@';
	}
	cout << "Nova rec: " << a << endl;
	return 0;
}
```

```
Unesi rec: komplikacija
Nova rec: komplik@cij@
```

Класа стринг

За обраду типа података стринг у C++, потребно је укључити библиотеку string.
Декларација промењивих типа података стринг:
string a, b, c;
Са овим типом података се могу извршавати следеће операције: ==, !=, <, >, >=, <=, =, спајање са другим стринговима...
Функције које омогућавају добијање дужине стринга (број знакова у стрингу): lenght(), size().
Пример:
string a(”0123456789”);
cout << a.length() << endl;
као резултат се добија 10 као дужина стринга a.

Приступ елементима стринга

Сваком знаку у оквиру стринга се може приступити помоћу оператора [] или функцијом at().
Пример: Одредити 6. знак у стрингу ”abcdefg”.
string s(”abcdefg”);
cout << s[5] << endl;
као резултат се добија знак f, пошто је индекс првог знака у стрингу 0.
Коришћењем функције at() се не мора водити рачуна о индексима јер ова функција проверава и да ли уопште постоји жељени индекс.
Пример:
string s(”abcdefg”);
cout << s.at(5) << endl;
као резултат се добија знак f.

Уметање знакова у стринг

За уметање знакова се користи функција insert().
Пример: У стринг ”aaaa” уметнути стрингове ”bbbb” и ”cccc”.
string s(”aaaa”);
cout << s << endl;
s.insert(2, string(”bbbb”));
cout << s << endl;
s.insert(4, ”cccc”);
cout << s << endl;
као резултат се добија: aabbccccbbaa

Пример01: а) Иницијализовати ниске знакова ”Dobar dan!” и ”Laku noc!” и приказати их на екрану.
б) Иницијализовати два стринга ”Kako ste ?” и ”Samo hrabro!” и приказати их на екрану.

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

int main()
{
	char niska_znakova_1[15] = "Dobar dan!";
	char niska_znakova_2[15] = "Laku noc!";
	string s_string_1("Kako ste ?");
	string s_string_2("Samo hrabro!");
	cout << "Niska znakova: " << niska_znakova_1 << endl;
	cout << "Niska znakova: " << niska_znakova_2 << endl;
	cout << "String: " << s_string_1 << endl;
	cout << "String: " << s_string_2 << endl;
	return 0;
}
```

Приступ елементима стринга

Пример01: Унети име и презиме ученика без и са коришћењем функције getline.

```
//Ispis imena ucenika klasa string
#include<iostream>
#include<string>
using namespace std;

int main()
{
	string ime, prezime;
	cout << "Unesi ime: ";
	cin >> ime;
	cout << "Unesi prezime: ";
	cin >> prezime;
	cout << ime << " " << prezime << endl;
	return 0;}
```

```
Unesi ime: Milan
Unesi prezime: Milanovic
Milan Milanovic
```

```
//Ispis imena ucenika klasa string sa getline
#include<iostream>
#include<string>
using namespace std;

int main()
{
	string ime;
	cout << "Unesi ime i prezime: ";
	getline(cin, ime);
	cout << ime << endl;
	return 0;
}
```

```
Unesi ime i prezime: Petar Petrovic
Petar Petrovic
```

Пример02: Одредити број знакова у унетом стрингу.

```
//Broj znakova u unetom stringu
#include<iostream>
#include<string>
using namespace std;

int main()
{
	string unos;
	cout << "Unesi rec: ";
	cin >> unos;
	cout << "Uneta rec ima duzinu " << unos.length() << endl;
	cout << "Uneta rec ima duzinu " << unos.size() << endl;
	return 0;
 }
```

```
Unesi rec: prestolonaslednik
Uneta rec ima duzinu 17
Uneta rec ima duzinu 17
```

```
Unesi rec: a kako ovo
Uneta rec ima duzinu 1
```

Пример03: У стрингу претвори слово а у @, без и са коришћењем функције at().

```
// Pretvaranje a u @
#include<iostream>
#include<string>
using namespace std;

int main()
{
	string unos;
	cout << "Unesi rec: ";
	cin >> unos;
	cout << "Uneta je rec " << unos << endl;
	for (int i = 0; i < unos.length(); i++)
	{
		if (unos[i] == 'a') unos[i] = '@';
	}
	cout << "Rec je prepravljena u " << unos << endl;
	return 0;
}
```

```
Unesi rec: katakomba
Uneta je rec katakomba
Rec je prepravljena u k@t@komb@
```

```
// Pretvaranje a u @ koriscenjem at()
#include<iostream>
#include<string>
using namespace std;

int main()
{
	string unos;
	cout << "Unesi rec: ";
	cin >> unos;
	cout << "Uneta je rec " << unos << endl;
	for (int i = 0; i < unos.length(); i++)
	{
		if (unos.at(i) == 'a') unos.at(i) = '@';
	}
	cout << "Rec je prepravljena u " << unos << endl;
	return 0;}
```

```
Unesi rec: ataman
Uneta je rec ataman
Rec je prepravljena u @t@m@n
```

<u>Рад са стринговима</u>

Пример01: На крају стринга додати нови стринг и 5 тачака.

```
// Upotrebom funkcije append dodati srtingu novi string i znakove
#include<iostream>
#include<string>
using namespace std;
int main()
{
	string s1("Hajde, ako si hrabar dodaj string ");
	string s2("i 5 tacaka stringu! ");
	string s3;
	s3.append(s1);
	s3.append(s2);
	s3.append(5, '.');
	cout << s3 << endl;
	return 0;
}
```

```
Hajde, ako si hrabar dodaj string i 5 tacaka stringu! .....
Press any key to continue . . .
```

Пример02: У стринг уметнути други стринг.

```
// Upotrebom funkcije insert umetnuti u string drugi string
#include<iostream>
#include<string>
using namespace std;

int main()
{
	string s1("Ovo je recenica.");
	string s2(" jedna");
	s1.insert(6, s2);
	cout << s1 << endl;
	return 0;
}
```

```
Ovo je jedna recenica.
Press any key to continue . . .
```

Пример03: У реченици избрисати једну реч.

```
// Upotrebom funkcije erase izbrisati rec iz recenice
#include<iostream>
#include<string>
using namespace std;

int main()
{
	string s1("Ovo je jedna recenica.");
	cout << "Ovo je nepromenjena recenica: ";
	cout << s1 << endl;
	s1.erase(7, 6);
	cout << "Ovo je promenjena recenica: ";
	cout << s1 << endl;
	return 0;
}
```

```
Ovo je nepromenjena recenica: Ovo je jedna recenica.
Ovo je promenjena recenica: Ovo je recenica.
```

Пример04: У реченици заменити једну реч другом.

```
// Upotrebom funkcije replace zameniti rec u recenici
#include<iostream>
#include<string>
using namespace std;

int main()
{
	string s1("Ovo je jedna recenica.");
	string s2("kratka");
	cout << "Ovo je nepromenjena recenica: ";
	cout << s1 << endl;
	s1.replace(7, 5, s2);
	cout << "Ovo je promenjena recenica: ";
	cout << s1 << endl;
	return 0;
}
```

```
Ovo je nepromenjena recenica: Ovo je jedna recenica.
Ovo je promenjena recenica: Ovo je kratka recenica.
```