

Текстуални низи во C++

Низи од знаци

C++ овозможува работа со текстуални податоци на неколку начини. Покрај со користење на класата `string` со која ќе се запознаеме подоцна, во C++ е дозволена работа со текстуални податоци и преку нивно третирање

Низи од знаци се креираат на ист начин како и останатите еднодимензионални низи:

```
char ime[N];
```

Притоа, `ime` го означува името на променливата, додека `N` го означува бројот на знаци (од колку елементи е составена низата `ime`). Еден од овие `N` знаци (зависно од текстуалниот податок кој ќе се чува во низата) ќе биде еднаков на `'\0'` - и ќе претставува т.н. `null` знак или `null terminator`. Овој знак е потребен за да го означи крајот на текстуалниот податок и да ни овозможи во низа со големина `N` да чуваме текстуални елементи кои се пократки од `N` знаци. На следната слика е претставен начинот на сместување на текстуални податоци во низата од знаци `name[13]`:

K	a	k	o		s	i	?	\0				
---	---	---	---	--	---	---	---	----	--	--	--	--

U	c	i	m	e		C	+	+	\0			
---	---	---	---	---	--	---	---	---	----	--	--	--

На овој начин, кога сакаме да печатиме податоци, програмата знае дека треба да печати знаци сè додека не стигне до знакот `'\0'`. Сите знаци по `'\0'` се неважни и не се користат.

Пр.

```
1char name[] = {'D', 'a', 'r', 'k', 'o', '\0'}; //treba da se navede '\0'  
2char name[] = "Darko"; //se podrazbira '\0'
```

И двата начина прикажани погоре креираат низа `name` со големина 6. Низата го содржи текстуалниот податок `"Darko"` и `null` знак `'\0'` за означување на крај на текстот. Бидејќи низите имаат големина која се дефинира при нивното креирање, не е дозволено подоцнежнo менување на нивната вредност преку користење на операторот `'='` (кај `string` класата ова е дозволено - таа работи со динамички резервирана меморија). На пример, следниов код е погрешен:

```
char name[] = "Darko";  
name = "Petar"; //GRESHKA!
```

Пр.

```
#include <iostream>  
using namespace std;  
int main()  
{  
char text[100];  
cout << "Vnesi eden zbor: ";  
cin >> text; //prochitaj eden zbor  
cout << text; //go pechati vneseniot zbor  
return 0;  
}
```

Во програмата дадена погоре, наредбата `cin >> text;` чита еден збор внесен од страна на корисникот и, истиот го сместува во променливата `text`. На крајот од текстот, автоматски, се додава и `null` знак. Бидејќи

cin чита податоци до првото појавување на празно место, tab или знак за нов ред, доколку сакаме да прочитаме цел ред текст (наместо само еден збор или број), потребно е да ја искористиме функцијата cin.getline(char[], N)

2. Функции за работа со стрингови:

Постојат 7 основни функции кои овозможуваат работа со текстуални низи (сите дефинирани во датотеката "<cstring>"):

- strlen(char[] niza) - ја враќа должината на низата niza (без null знак)
- strcpy(destinacija, izvor) - копира текстуална низа (од izvor во destinacija), вклучувајќи го и null знакот. Внимавајте: низата destinacija треба да има доволна големина за да ги собере сите знаци од izvor.
- strncpy(destinacija, izvor, N) - копира најмногу N знаци од текстуална низа (од izvor во destinacija). Знакот '\0' ќе се ископира само доколку се појави во првите N знаци - инаку, истиот мора да го додадеме самите. Внимавајте: низата destinacija треба да има доволна големина за да ги собере сите потребни знаци.
- strcmp(prva, vtor) - споредува две низи (резултатот е 0 ако низите се еднакви, инаку е 1 и променливата во која се сместува резултатот се декларира како логичка променлива)
- strncmp(prva, vtor, int N) - споредува N знаци (или помалку, ако '\0' се појави претходно) од низите prva и vtor (резултатот е 0 ако првите N знаци од низите се еднакви)
- strcat(prva, vtor) - ја надоврзува низата vtor на prva. Внимавајте: низата prva треба да има доволна големина за да ги собере знаците од двете низи.
- strncat(prva, vtor, N) - ги надоврзува првите N знаци (или помалку, ако '\0' се појави претходно) од низата vtor на prva. Внимавајте: низата prva треба да има доволна големина за да ги собере сите потребни знаци.

3. Функции за работа со знаци:

- isalpha(ch) – true ако ch е алфабетски знак инаку false
- isupper(ch)- true ако ch е голема буква инаку false
- islower(ch)- true ако ch е мала буква инаку false
- isdigit(ch)- true ако ch е цифра од 0 до 9 инаку false
- toupper(ch), tolower(ch) – претвораат во мала т.е. во голема буква

Пр. Програма со која даден текст внесен преку тастатура се печати .

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    char tekst[100]; //se deklarira promenliva koja se vika tekst i moze da sodrzi najmnogu 100
                    //char promenlivi, odnosno najmnogu 100 znaci (karakter)

    int i,dolzina;
    cout << "Vnesi tekst: ";
    cin.getline(tekst,100); //cin.getline go cita tekstot sto se vnesuva preku tastatura i go smestuva
                            //vo promenlivata tekst

    dolzina=strlen(tekst); //strlen ja naogja dolzinata na promenlivata tekst,
    cout<<"dolinata na tekstot e"<<dolzina<<endl; //odnosno brojot na karakteri
    cout<<"Go pecatam:"<<endl;
    for(i=0;i<=dolzina-1;i++) cout << tekst[i]; //gi pecati karakterite ocnuvajki od prviot pa do posledniot
    return 0;
}
```

Пр. Програма со која даден внесен текст преку тастатура се печати во обратен редослед.

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    char tekst[100];                //se deklarira promenлива koja se vika tekst i moze da sodrzi najmnogu 100
                                   //char promenlivi, odnosno najmnogu 100 znaci (karakteri)

    int i,dolzina;
    cout << "Vnesi tekst: ";
    cin.getline(tekst,100);        //cin.getline go cita tekstot sto se vnesuva preku tastatura i go smestuva
                                   //vo promenlivata tekst
    dolzina=strlen(tekst);         //strlen ja naogja dolzinata na promenlivata tekst,
                                   //odnosno brojot na karakteri
    cout<<"dolinata na tekstot e"<<dolzina<<endl;
    cout<<"Go pecatam:"<<endl;
    for(i=dolzina-1;i>=0;i--) cout << tekst[i];    //gi pecati karakterite ocnuvajki od prviot pa do posledniot
    return 0;
}
```

Пр. Програма со која се проверува дали два внесени текста се еднакви или не.

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    char tekst1[100],tekst[100];
    int i,dolzina;
    bool t;
    cout << "Vnesi tekst: ";
    cin.getline(tekst,100);
    cin.getline(tekst1,100);
    t=strcmp(tekst, tekst1);
    if(t==0) cout<<"Tekstovite se isti";
    else cout<<"tekstovite ne se isti";
    return 0;
}
```

Пр. Програма со која се проверува дали внесениот текст е палиндром или не. (Палиндром е текст кој се чита исто во двете насоки – од лево кон десно и од десно кон лево)

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
    char tekst[100],tekst_obratno[100];
    bool palindrom;
    int i,dolzina;
    cout << "Vnesi tekst: ";
    cin.getline(tekst,100);
```

```

dolzina=strlen(tekst);

for(int i=dolzina-1;i>=0;i--)
    tekst_obratno[i]=tekst[dolzina-i-1];

for(int i=0;i<=dolzina-1;i++)
{
    if (tekst[i]==tekst_obratno[i])
        palindrom=true;
    else
    {
        palindrom=false;
        cout<<"Vneseniot tekst ne e palindrom"<<endl;
        break;
    }
}

    if (palindrom==true)
        cout<<"Vneseniot tekst e palindrom"<<endl;
return 0;
}

```

Пр. Програма што ќе избори и печати колку мали самогласки има во една реченица внесена од тастатура

```

#include<iostream>
#include<cstring>
using namespace std;
int main()
{
char tekst[100];
bool palindrom;
int b=0,dolzina;
cout << "Vnesi tekst: ";
cin.getline(tekst,100);
dolzina=strlen(tekst);

for(int i=dolzina-1;i>=0;i--)
{
    if (tekst[i]=='a' || tekst[i]=='o' || tekst[i]=='e' || tekst[i]=='i' ||tekst[i]=='u')
        b++;
    }
    cout<<"Vneseniot tekst ima "<<b<< "samoglaski"<<endl;
return 0;
}

```

Пр. програма со која се определува колку знаци во дадена низа знаци се букви.

```

#include <iostream>
#include <cstring>

```

```

using namespace std;
int main()
{
    char a[100];
    int d,i,b=0;
    cout<<"Vnesi niza "<<endl;
    cin.getline(a,100);
    d=strlen(a);
    for(i=0;i<=d-1;i++)
        if (isalpha(a[i])) b++;

    cout<<"ima "<<b;
    return 0;
}

```

Пр. Програма со која сите букви во дадена низа знаци се претвораат во мали.

```

#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char a[100];
    int d,i;
    cout<<"Vnesi niza "<<endl;
    cin.getline(a,100);
    d=strlen(a);
    for(i=0;i<=d-1;i++)
        a[i]=tolower(a[i]);
    for(i=0;i<=d-1;i++)
        cout<<a[i];
    return 0;
}

```

Пр. програма со која од дадена низа знаци се печатат само буквите

```

#include<iostream>
#include<cstring>
using namespace std;
int main(){
char tekst[100];

int i,dolzina;
cout << "Vnesi tekst: ";
cin.getline(tekst,100);
dolzina=strlen(tekst);
for(int i=0;i<=dolzina-1;i++)
{
    if (((tekst[i]>='a'&& tekst[i]<='z')) || (((tekst[i]>='A'&& tekst[i]<='Z'))))
        cout<<tekst[i];
}
}

```

```
return 0;
}
```

Или:

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
char tekst[100],a[100];

int i,dolzina,j=0;
cout << "Vnesi tekst: ";
cin.getline(tekst,100);
dolzina=strlen(tekst);
for(int i=0;i<=dolzina-1;i++)
{
if (((tekst[i]>='a'&& tekst[i]<='z')) || ((tekst[i]>='A'&& tekst[i]<='Z')))
{
a[j]=tekst[i];
j++;
}
}
for(i=0;i<j;i++)
cout<<a[i];

return 0;
}
```