

Polja znakova

**Rukovanje tekstualnim informacijama
upotrebom tzv. C-stringova**

C stringovi

■ C i C++

- nizovi znakova spremljeni u polja znakova (u pojedini član polja spremljen je po jedan znak niza)
- za rukovanje koristimo
adresu prvog slova (znaka) u stringu
varijable tipa pokazivač
odgovarajuće funkcije iz biblioteke

■ C++

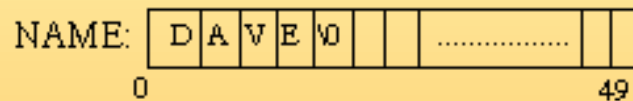
- ima i tzv. klasu (razred) `string` (noviji - napredniji način)

■ C-stringovi brži u izvođenju od rukovanja stringovima iz klase `string`

Značajke

- Za čuvanje nizova znakova deklariramo polja znakova
- Najveći broj znakova koje se u polje može smjestiti **za jedan je manji od broja u uglatim zagradama**
- Ime polja bez zagrada znači adresu (ne može se pisati s lijeve strane znaka pridruživanja)

```
char NAME[50];
```



Primjer 1. Brojanje slova i znamenki u nizu znakova

```
#include <iostream>
#include <stdio.h>
#include <ctype.h>
#define MAX 100
using namespace std;

int main(){
    char niz[MAX];
    int i, s=0, b=0;
    cout << "Unesite string: ";
    gets(niz);
    for (i=0;niz[i]!='\0';i++)
        if (isalpha(niz[i])) s++;
        else
            if(isdigit(niz[i])) b++;
    cout << "Niz sadrzi " << s << " slova.\n";
    cout << "Niz sadrzi " << b << " brojeva.\n";
    return 0;}
```

Primjeri

```
strlwr (niz1) ;  
    //ugrađena funkcija; mijenja sva slova niza u mala  
if (strlen (niz1) ==strlen (niz2) )  
    //funkcija i usporedba (dužina niza1 i niza2)  
if (niz1 [i] !=niz2 [i] )  
    //usporedba članova niza na istim pozicijama
```

Nizovne funkcije C++ biblioteke

isalnum	Vraća true ako je argument slovo ili znamenka
isalpha	Vraća true ako je argument slovo
isdigit	Vraća true ako je argument znamenka
islower	Vraća true ako je argument malo tiskano slovo
isupper	Vraća true ako je argument veliko tiskano slovo
tolower	Ako je argument veliko tiskano slovo vraća odgovarajuće malo
toupper	Ako je argument malo tiskano slovo vraća odgovarajuće veliko

Nizovne funkcije za pretvorbu iz niza znakova u numeričku vrijednost

atoi	Nalazi prvi cijeli broj u stringu i vraća vrijednost tog broja (int).
atof	Nalazi prvi realni broj u stringu i vraća vrijednost tog broja (double).

Kopiranje stringa ili dijela stringa u memoriju rezerviranu za drugi string

strcat	Kopira drugi string na kraj prvog stringa i vraća adresu prvog.
strcpy	Kopira drugi string na početak prvog stringa i vraća adresu prvog.
strncpy	Kopira određeni broj znakova iz drugog stringa na početak prvog stringa i vraća adresu prvog stringa.
strtok	Kopira null znak u prvi string na lokaciju podudarajućeg znaka u drugom stringu i vraća adresu lokacije na kojoj je smješten null znak.

Nalaženje adrese ili pozicije određenog znaka, stringa ili dijela stringa u danom nizu znakova

strchr	Pronalazi određeni pojedinačni znak u stringu i vraća adresu prve pojave znaka.
strcspn	Pronalazi prvu pojavu u prvom stringu bilo kojeg znaka u drugom stringu i vraća poziciju znaka u prvom stringu.
strpbrk	Pronalazi prvu pojavu u prvom stringu bilo kojeg znaka u drugom stringu i vraća adresu znaka u prvom stringu.
strrchr	Pronalazi posljednju pojavu u prvom stringu bilo kojeg znaka u drugom stringu i vraća adresu znaka u prvom stringu.
strspn	Pronalazi prvi znak u prvom stringu koji se ne pojavljuje u drugom stringu i vraća poziciju znaka u prvom stringu.
strstr	Pronalazi prvu pojavu drugog stringa u prvom stringu i vraća adresu početka drugog stringa u prvom stringu.

Leksikografska usporedba stringova, nalaženje dužine stringa, određivanje adrese stringa koji opisuje pogrešku

strcmp	Uspoređuje prvi string sa drugim stringom i vraća int koji pokazuje koji je string leksikografski veći.
strncmp	Uspoređuje specificirani broj znakova iz prvog stringa sa drugim stringom i vraća int koji pokazuje koji je string leksikografski veći.
strlen	Vraća broj znakova u stringu (bez null znaka) → koliko je znakova stvarno spremljeno u string.

Primjer 3.

```
#include <iostream>
#include <stdio.h>
#define MAX_NIZ 10
#define MAX_ZNAK 50

using namespace std;

int main(){
    char nizovi[MAX_NIZ][MAX_ZNAK];
    int n, i, j;
    cout << "Unesite N (max. 10): "; cin >> n;

    for (i=0;i<n;i++){
        cout << "Unesite " << (i+1) << ". niz: "; gets(nizovi[i]);}
    for (i=0;i<n;i++){
        for (j=0;nizovi[i][j]!='\0';j++)
            if(nizovi[i][j]!=' ')
                cout << nizovi[i][j];
        cout << endl;}

    return 0;}
```

Primjer 4.

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

int main() {
    char aa;
    char bb[4], cc[100], dd[100];
    aa = 'g'; bb[0] = 'C'; bb[1] = 'a'; bb[2] = 't'; bb[3] = '\\0';
    // Posljednji znak polja popunjava se sa \\0 (nul-znak)-bajt sa svim bitovima u 0
    strcpy (cc, " Red 1. \\n Red 2.");
    //način inicijalizacije stringa-kopira string u polje i automatski dodaje završni \\0
    strcpy (dd, cc); //kopira sadržaj iz cc u dd, ne smije se koristiti dd=cc
    cout << aa << endl << bb << endl << cc << endl << dd << endl;
    return 0; } //za numerička polja ispisala bi se adresa početka polja, a ne sadržaj
```

- Funkciji `strcpy` predaju se adrese dva polja – kopija sadržaja drugog, sprema se na adresi prvog polja u listi argumenata (mora biti dosta veliko za sve članove)

Upotreba funkcije `strlen`

Kopiranje sadržaja polja

- `strlen` - prenosimo adresu početka svakog stringa (retka):
 - `strlen(bb[0])` prenosi funkciji `strlen` adresu početka prvog retka od `bb[][]`
 - `strlen` vraća broj znakova u tom retku (bez null znaka)
- `strcpy` - kopira iz jednog znakovnog polja u drugo, npr. za dva ista polja:
 - ```
for (i=0; i<4; i++) strcpy (cc[i], dd[i]);
```

  
//4 retka, kopira sadržaj iz `dd[][]`, redak po redak u `cc[][]`
- Za čitanje više od jedne riječi: objekt `cin` i neki funkcijski član

# C Stringovi (znakovna polja)

- Znakovno polje (niz) – polje čije su komponente tipa `char`
- String – niz od nula ili više znakova ograničen sa dvostrukim navodnicima
- C stringovi završavaju nul-znakom (*null terminated* (`'\0'`))
- Posljednji znak u nizu je tzv. `null character`

# C Strings (Character Arrays) (continued)

- Razlika je između 'A' i "A"
  - 'A' je znak A
  - "A" je string A
- "A" predstavlja dva znaka, 'A' i '\0'
- "Hello" sadrži 6 znakova 'H', 'e', 'l', 'l', 'o', i '\0'

```
char ime[16];
```

- Najveći string koji se može spremiti u ime je 15

# C Stringovi

```
char ime[16] = "Jurica";
```

deklarira string varijablu `ime` dužine 16 i sprema u nju  
`"Jurica"`

```
char ime[] = "Jurica";
```

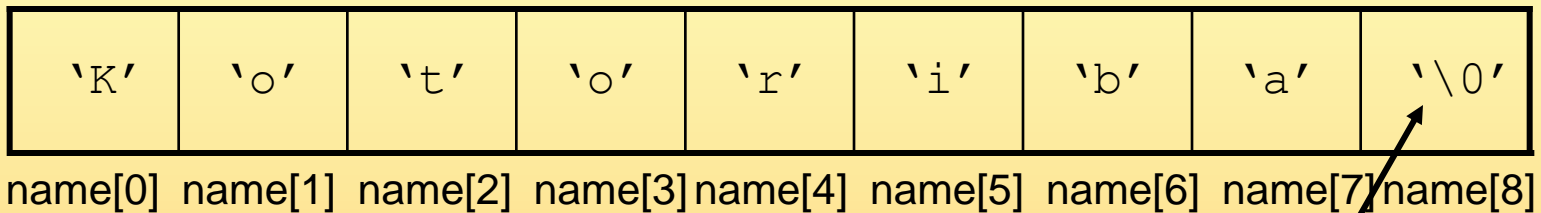
deklarira string varijablu `ime` dužine 7 i sprema u nju  
`"Jurica"`



# Inicijalizacija znakovnog polja

- Stavljanjem string u dvostruke navodnike

```
char name[9]="Kotoriba";
```



String literals are *null terminated* automatically in C++.

# Druge inicijalizacije polja w/ Strings

```
char name[] = "Mark";
```

```
char name[5] =
 { 'M', 'a', 'r', 'k', '\0' };
```

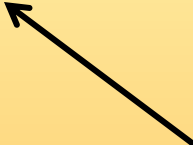
```
char name[] =
 { 'M', 'a', 'r', 'k', '\0' };
```

# Znakovna polja

- Ispis stringa


```
char name[] = "Mark";
cout << name;
```

Može ispisati sadržaj znakovnog polja u jednoj naredbi



# Čitanje stringa

```
char name[25];
cout << "Please enter your name: ";
cin >> name;
```

- 
- Prihvaća samo do prvog razmaka.
  - automatski dodaje null znak.

**TABLE 9-1** strcpy, strcmp, and strlen functions

| Function                    | Effect                                                                                                                           |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <code>strcpy(s1, s2)</code> | Copies the string s2 into the string variable s1<br>The length of s1 should be at least as large as s2                           |
| <code>strcmp(s1, s2)</code> | Returns a value < 0 if s1 is less than s2<br>Returns 0 if s1 and s2 are the same<br>Returns a value > 0 if s1 is greater than s2 |
| <code>strlen(s)</code>      | Returns the length of the string s, excluding the null character                                                                 |

# Usporedba stringova

- **C-stringovi se uspoređuju znak po znak**
- **Za ASCII skup znakova**
  1. `string "Air"` je manji od stringa `"Boat"`
  2. `string "Air"` je manji od stringa `"An"`
  3. `string "Bill"` je manji od stringa `"Billy"`
  4. `string "Hello"` je manji od `"hello"`

## EXAMPLE 9-8

```
char studentName[21];
char myname[16];
char yourname[16];
```

### Statement

```
strcpy(myname, "John Robinson");
```

```
strlen("John Robinson");
```

```
int len;
```

```
len = strlen("Sunny Day");
```

```
strcpy(yourname, "Lisa Miller");
```

```
strcpy(studentName, yourname);
```

```
strcmp("Bill", "Lisa");
```

```
strcpy(yourname, "Kathy Brown");
```

```
strcpy(myname, "Mark G. Clark");
```

```
strcmp(myname, yourname);
```

### Effect

Myname = "John Robinson"

Returns 13, the length of the string  
"John Robinson"

Stores 9 into len

yourname = "Lisa Miller"  
studentName = "Lisa Miller"

Returns a value < 0

yourname = "Kathy Brown"  
myname = "Mark G. Clark"  
Returns a value > 0

# Čitanje i pisanje stringova

## ■ Ulaz

- `cin >> name;` sprema slijedeći ulazni string u `name`
- Stringovi sa prazninama ne mogu biti čitani upotrebom operatora `>>`



# Čitanje i pisanje stringova (nastavak)

- Za čitanje stringova sa prazninama, koristi se funkcija `get` sa `cin`:

```
cin.get(str, m+1);
```

Sprema slijedećih  $m$  znakova u `str`, ali se newline znak ne sprema u `str`

Ako ulazni string ima manje od  $m$  znakova, čitanje prestaje kod newline znaka

# Čitanje i pisanje stringova (nastavak)

## ■ Ispis stringa

- naredba `cout << name;` ispisuje sadržaj varijable `name` na ekranu
- operator `<<` nastavlja sa ispisom sadržaja od `name` dok ne naiđe na null znak
- ako `name` ne sadrži null znak, `<<` nastavlja ispisivati podatke iz memorije susjedne `name` dok se ne naiđe na `'\0'`