
Znakovni nizovi

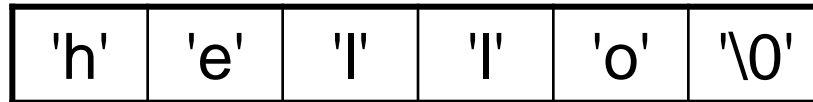
Stringovi

Pojam stringa

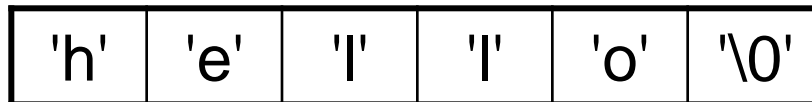
```
char niz_znakova[5] = {'h', 'e', 'l', 'l', 'o'};
```



```
char string[6] = {'h', 'e', 'l', 'l', 'o', '\0'};
```



```
char string[ ] = "hello";
```



Primjer: Ponavljanje iz Prog1

```
#include <stdio.h>
#include <string.h>

int main( ){
    char ime_i_prezime[80];

    scanf("%s", ime_i_prezime);

    gets(ime_i_prezime);
    puts(ime_i_prezime);          /* '\0' → '\n' */
```

Primjer: Ponavljanje iz Prog1 (2)

```
printf("%d\n", strlen(ime_i_prezime));
```

```
printf("%c\n", ime_i_prezime[4]);
```

```
printf("%d\n", sizeof("12 34\n\t"));
```

```
printf("%d\n", strlen("12 34\n\t"));
```

```
return 0;
```

```
}
```

Pokazivači i stringovi

```
char polje[10], *ptr;
```

```
polje = "hello";          /* netočno */
```

```
ptr = "hello";           /* točno */
```

```
printf ("%s", ptr);      /* ispis stringa */
```

```
printf ("%u", ptr);      /* ispis adrese – npr. 1156 */
```

'h'	'e'	'l'	'l'	'o'	'\0'
-----	-----	-----	-----	-----	------

1156 1157

1161

Primjer:

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char *ptr = "hello";

    printf("%c\n", *ptr);
    printf("%s\n", ptr);

    printf("%c\n", *ptr + 2);
    printf("%s\n", ptr + 2);

    return 0;
}
```

Primjer:

```
#include <stdio.h>
#include <string.h>

void main(void){
    char string[80];
    char temp, *p, *k;
    gets (string);
    p = string;
    k = p + (strlen(string) - 1);
    while (k > p){
        temp = *p;
        *p++ = *k;
        *k-- = temp;
    }
    printf ("%s\n", string);
}
```

Neke funkcije za rad sa stringovima

```
#include <string.h>
size_t strlen(const char *);
```

Primjer: Moguće implementacije funkcije `strlen`.

```
int strlen(register char *s)
{
    int n = 0;

    while (*s++)
        ++n;
    return(n);
}
```

```
int strlen(char *s)
{
    char *p = s;

    while (*p != '\0')
        p++;
    return(p - s);
}
```

Neke funkcije za... (2)

Primjer: Moguća implementacija funkcije `strlen`.

```
int strlen(char s[])
{
    int i;

    i = 0;

    while (s[i] != '\0')
        ++i;
    return i;
}
```

Neke funkcije... (3)

```
char *strcat(char *dest, const char *src);  
char *strchr(const char *, int);  
char *strcpy(char *, const char *);  
int strcmp(const char *, const char *);  
.....
```

```
char *strcat(char *dest, char *src) /* moguća implementacija */  
{  
    char *p;  
    p = dest;  
    while (*p++);  
    --p;  
    while (*p++ = *src++);  
    return(dest);  
}
```

Funkcija `strcat`

Primjer: Moguća implementacija funkcije `strcat`.

```
void strcat(char s[], char t[])
{
    int i, j;

    i = j = 0;

    while (s[i] != '\0')
        i++;
    while ((s[i++] = t[j++]) != '\0')
        ;
}
```

Funkcija `strcmp`:

```
char foo[] = "hello";  
char bar[] = "hello";
```

```
if (foo == bar)  
    puts("Stringovi su jednaki.");  
else  
    puts("Stringovi nisu jednaki.");
```

```
if (!strcmp(foo, bar))  
    puts("Stringovi su jednaki.");  
else  
    puts("Stringovi nisu jednaki.");
```

Primjer:

```
#include<stdio.h>
#include<string.h>

int main( ){
    int i;
    char alpha[30], chr[2] = " ";
    strcpy(alpha, "a");
    for(i = 'b'; i <= 'z'; i++){
        chr[0] = i;
        strcat(alpha, chr);
    }
    printf("%s\n", alpha);
    return 0;
}
```

Neke funkcije za testiranje znakova

```
#include <ctype.h>
int isalnum(int c);
int isalpha(int c);
int isdigit(int c);
.....
```

Primjer: moguće implementacije funkcije `isdigit`.

```
#define isdigit(x) ((x) >= '0' && (x) <= '9')
.....
printf("%d\n", isdigit('0'));          /* 1 */
printf("%d\n", isdigit('C'));          /* 0 */
```

Moguće implementacije

```
int isdigit(int x)
```

```
{
```

```
    if('0' <= x && x <= '9') return 1;
```

```
    else return 0;
```

```
}
```

```
.....
```

```
if('A' <= c && c <= 'Z' || 'a' <= c && c <= 'z') return 1;
```

```
else return 0;
```

Primjeri nekih funkcija

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main( ){
    int i; char kolegij[]="Programiranje 2";

    puts(strupr(kolegij));

    for(i = 0; kolegij[i] != '\0'; i++)
        if(islower(kolegij[i]))
            kolegij[i] = toupper(kolegij[i]);
    puts(kolegij);

    return 0;}
```

Primjer: Broj riječi u stringu

```
#include <stdio.h>
#include <string.h>

#define TRUE 1
#define FALSE 0

int broj_rijeci(char* str)
{
    int brojac = 0;
    int razmak = TRUE;
```

Primjer: Broj riječi u stringu (2)

```
while (*str != NULL){
    if ((*str == ' ') || (*str == '\t')){
        if (!razmak){
            brojac++;
            razmak = TRUE;
        }
    }
    else
        razmak = FALSE;
    str++;
}

if (!razmak) brojac++;
return brojac;
}
```