

Programiranje u fizici

13. Rad sa stringovima

Prirodno-matematički fakultet u Nišu
Departman za fiziku

- Već smo spomenuli da u programskom jeziku C postoji poseban tip podataka za rad sa znakovima – karakterima:

```
char c;
```

- Postavlja se pitanje šta se dešava sa nizom znakova?
- U programskom jeziku C nemamo poseban tip za niz znakova koga često nazivamo string.
- Zato za smeštanje stringa se koristi niz tj. niz karaktera.

```
char pozdrav[10];
```

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

```
int main()  
{  
    char pozdrav [10];  
  
    pozdrav [0] = 'z';  
    pozdrav [1] = 'd';  
    pozdrav [2] = 'r';  
    pozdrav [3] = 'a';  
    pozdrav [4] = 'v';  
    pozdrav [5] = 'o';  
  
    printf("%s", pozdrav);  
  
    return 0;
```

} 11:01

11:08

Zdravo@
Process returned 0 (0x0) execution time : 0.025 s
Press any key to continue.

Zdravoü
Process returned 0 (0x0) execution time : 0.025 s
Press any key to continue.

0	1	2	3	4	5	6	7	8	9
Z	d	r	a	v	o	@	?	?	?

slučajne vrednosti

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

```
int main()  
{
```

```
char pozdrav [10];
```

```
pozdrav [0] = 'z';
```

```
pozdrav [1] = 'd';
```

```
pozdrav [2] = 'r';
```

```
pozdrav [3] = 'a';
```

```
pozdrav [4] = 'v';
```

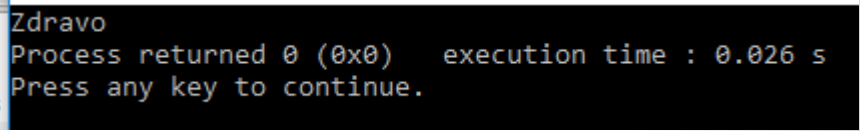
```
pozdrav [5] = 'o';
```

```
pozdrav [6] = '\\0';
```

```
printf ("%s", pozdrav);
```

```
return 0;
```

String – niz karaktera u programskom jeziku C u suštini predstavlja jedno-dimenzionalni niz karaktera koji se završava tkz. null karakterom '\\0'.



0	1	2	3	4	5	6	7	8	9
Z	d	r	a	v	o	\\0	?	?	?

slučajne vrednosti

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

```
int main()  
{
```

```
    char pozdrav [10];
```

```
    pozdrav [0] = 'z';
```

```
    pozdrav [1] = 'd';
```

```
    pozdrav [2] = 'r';
```

```
    pozdrav [3] = 'a';
```

```
    pozdrav [4] = 'v';
```

```
    pozdrav [5] = 'o';
```

```
    pozdrav [3] = '\\0';
```

```
    printf("%s", pozdrav);
```

```
    return 0;
```

```
Zdr  
Process returned 0 (0x0)   execution time : 0.025 s  
Press any key to continue.
```

0	1	2	3	4	5	6	7	8	9
Z	d	r	\\0	v	o	?	?	?	?

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

int main()
{
    char pozdrav[7] = "Zdravo";

    printf ("%s\n", pozdrav);

    return 0;
}
```

- u ovom slučaju nije potrebno da programer dodaje null karakter na kraju znakovne konstante.
- Sam C prevodilac će automatski dodati null karakter na kraju niza u postupku inicijalizacije istog.

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

min dimenzija niza = broj karaktera + 1

```
int main()
```

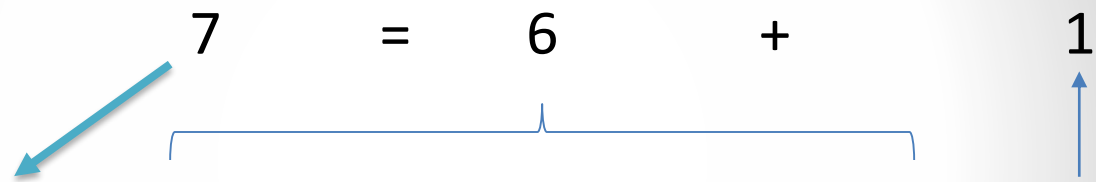
```
{
```

```
char pozdrav[7] = {'z', 'd', 'r', 'a', 'v', 'o', '\0'};
```

```
printf ("%s\n", pozdrav);
```

```
return 0;
```

```
}
```



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

void print_str (char* s) {
    int i=0;

    while (s[i] != '\0') {
        printf("%c", s[i]);
        i++;
    }
    printf("\n");
}
```

```
int main()
{
    char rec[20] = "";
    printf("Unesi rec: ");
    scanf("%s", rec);
    print_str (rec);
    return 0;
}
```

```
Unesi rec: Pozdrav
Pozdrav

Process returned 0 (0x0)   execution time : 2.774 s
Press any key to continue.
```

0	1	2	3	4	5	6	7	8	...		19
P	o	z	s	r	a	v	\0	?	...		?

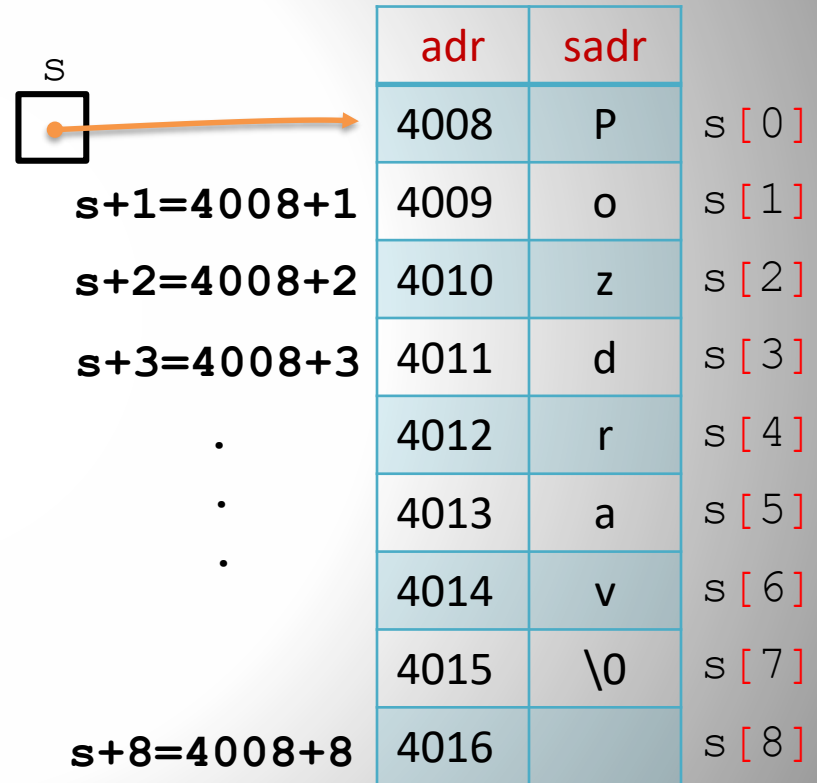

```
#include <stdio.h>
#include <stdlib.h>
```

```
void print_str (char* s) {
    int i=0;

    while (s[i] != '\0') {
        printf("%c", s[i]);
        i++;
    }
    printf("\n");
}
```

```
int main()
{
    char rec[20] = "";
    printf("Unesi rec: ");
    scanf("%s", rec);
    print_str (rec);
    return 0;
}
```

Prenos "stringa" u f-ciju UVEK po reference.
 Zašto?



0	1	2	3	4	5	6	7	8	...	19
P	o	z	s	r	a	v	\n	?	...	?

Programski jezik C ima niz funkcija za rad sa stringovima.

Funkcija	Objašnjenje
<code>gets (s1);</code>	Prihvata uneti niz karaktera sa tastature i smešta ih u s1
<code>puts (s1);</code>	Štampa na ekran vrednost stringa s1
<code>strcpy (s1, s2);</code>	Kopira string s2 u string s1
<code>strcat (s1, s2);</code>	Nadovezuje string s2 na kraj stringa s1
<code>strlen (s1);</code>	Vraća dužinu stringa s1 u karakterima
<code>strcmp (s1, s2);</code>	Vraća 0 ako su s1 i s2 identični stringovi; manje od 0 ako je s1 < s2; veće od 0 ako je s1 > s2
<code>strchr (s1, ch);</code>	Vraća pokazivač na prvo pojavljivanje karaktera ch u stringu s1
<code>strstr (s1, s2);</code>	Vraća pokazivač na prvo pojavljivanje stringa s2 u stringu s1

Kompletna lista funkcija za rad sa stringovima može se naći u biblioteci [string.h](#)

Tabele, vektori i matrice
Funkcije
Rad sa stringovima
Rad sa strukturama
Rad sa datotekama

Označavanje kraja string-a
Rezervacija memorije za string
Štampanje stringa – korisnička f-cija
Prenos stringa u f-ciju
Funkcije za rad sa stringovima – gets ()
Argumenti komandne linije

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

```
int main()  
{  
    char rec [100];  
  
    printf("Unesite neku rec:\n");  
  
    gets(rec);  
  
    printf("Uneta rec je: %s", rec);  
  
    return 0;  
}
```

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

```
int main()  
{  
    char rec [100];  
  
    printf("Unesite neku rec:\n");  
  
    scanf("%s", rec);  
  
    printf("Uneta rec je: %s", rec);  
  
    return 0;  
}
```

```
Unesite neku rec:  
Programiranje u fizici.  
Uneta rec je: Programiranje u fizici.  
Process returned 0 (0x0)   execution time : 6.438 s  
Press any key to continue.
```

```
Unesite neku rec:  
Programiranje u fizici  
Uneta rec je: Programiranje ?  
Process returned 0 (0x0)   execution time : 5.762 s  
Press any key to continue.
```

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

int main()
{
    char rec [30];

    printf("Unesite neku rec:\n");

    gets (rec);

    puts ("Uneta rec je:");
    puts (rec);

    return 0;
}
```

```
Unesite neku rec:
Programiranje u fizici
Uneta rec je:
Programiranje u fizici

Process returned 0 (0x0)   execution time : 6.133 s
Press any key to continue.
```

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

int main()
{
    char rec [30];
    char poruka [45] = "Uneta rec je: "; // 15 + 30

    printf("Unesite neku rec:\n");

    gets (rec);

    puts (strcat (poruka, rec));

    return 0;
}
```

```
Unesite neku rec:
Programiranje u fizici
Uneta rec je: Programiranje u fizici
Process returned 0 (0x0)   execution time : 6.521 s
Press any key to continue.
```

```
#include <stdio.h>
#include <string.h>
int main ()
{
    char str1[6] = "Hello";
    char str2[6] = "World";
    char str3[12];
    int len ;

    /* kopira str1 u str3 */
    strcpy(str3, str1);
    printf("strcpy (str3, str1): %s\n", str3);

    /* spaja str1 i str2 */
    strcat( str1, str2);
    printf("strcat (str1, str2): %s\n", str1);

    /* ukupna duzina str1 posle spajanja */
    len = strlen(str1);
    printf("strlen (str1) : %d\n", len);

    return 0;
}
```

```
strcpy (str3, str1): Hello
strcat (str1, str2): HelloWorld
strlen (str1) : 10

Process returned 0 (0x0)   execution time : 0.024 s
Press any key to continue.
```

```
#include <stdio.h>  
#define MAX_SIZE 100
```

```
void izbaci_praznine(char* s, char* rez) {  
    while(*s) {  
        if (*s==' ' || *s=='\t') {  
            s++;  
        }  
        else {  
            *rez++=*s++;  
        }  
    }  
  
    *rez='\0';  
}
```

while (*s != '\0') {
while (*s != 0) {

```
int main()  
{  
    char rez[MAX_SIZE], str[MAX_SIZE];  
  
    printf("Unesi string:");  
  
    if (gets(str)==NULL) {  
        return 1;  
    }  
  
    izbaci_praznine(str, rez);  
  
    printf("%s\n", rez);  
  
    return 0;  
}
```

```
Unesi string:Programiranje u fizici  
Programiranjeufizici  
Process returned 0 (0x0)   execution time : 8.218 s  
Press any key to continue.
```

```
#include <stdio.h>
#define MAX_SIZE 100

void izbacij_praznine(char* s, char* rez) {
    while(*s) {
        if (*s==' ' || *s=='\t') {
            s++;
        }
        else {
            *rez++=*s++;
        }
    }
    *rez='\0';
}

int main()
{
    char rez[MAX_SIZE], str[MAX_SIZE];

    printf("Unesi string:");

    if (gets(str)==NULL) {
        return 1;
    }

    izbacij_praznine(str, rez);

    printf("%s\n", rez);

    return 0;
}
```

adr	sadr
4008	P
4009	r
4010	o
4011	g
4012	r
4013	a
4014	m
4015	i
4016	r

s[0]
s[1]
s[2]
s[3]
s[4]
s[5]
s[6]
s[7]
s[8]

adr	sadr
7001	P
7002	r
7003	o
7004	g
7005	r
7006	a
7007	m
7008	i
7009	r

rez[0]
rez[1]
rez[2]
rez[3]
rez[4]
rez[5]
rez[6]
rez[7]
rez[8]

```
Unesi string:Programiranje u fizici
Programiranjeufizici

Process returned 0 (0x0)   execution time : 8.218 s
Press any key to continue.
```



```
#include <stdio.h>
#include <string.h>
int main()
{
    char str1[100], str2[100], str[200];
    char* s1, *s2, *s;
    int i=0;

    printf("Unesite prvi string: ");
    gets(str1);

    printf("Unesite drugi string: ");
    gets(str2);

    s1=str1;
    s2=str2;
    s=str;

    strcpy (s, str1);
    strcpy (s+strlen(str1), str2);

    printf("Rezultujući string je: %s", str);
    return 0;
}
```

```
Unesite prvi string: Programiranje
Unesite drugi string: fizika
Rezultujući string je: Programiranjefizika
Process returned 0 (0x0)   execution time : 7.474 s
Press any key to continue.
```

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

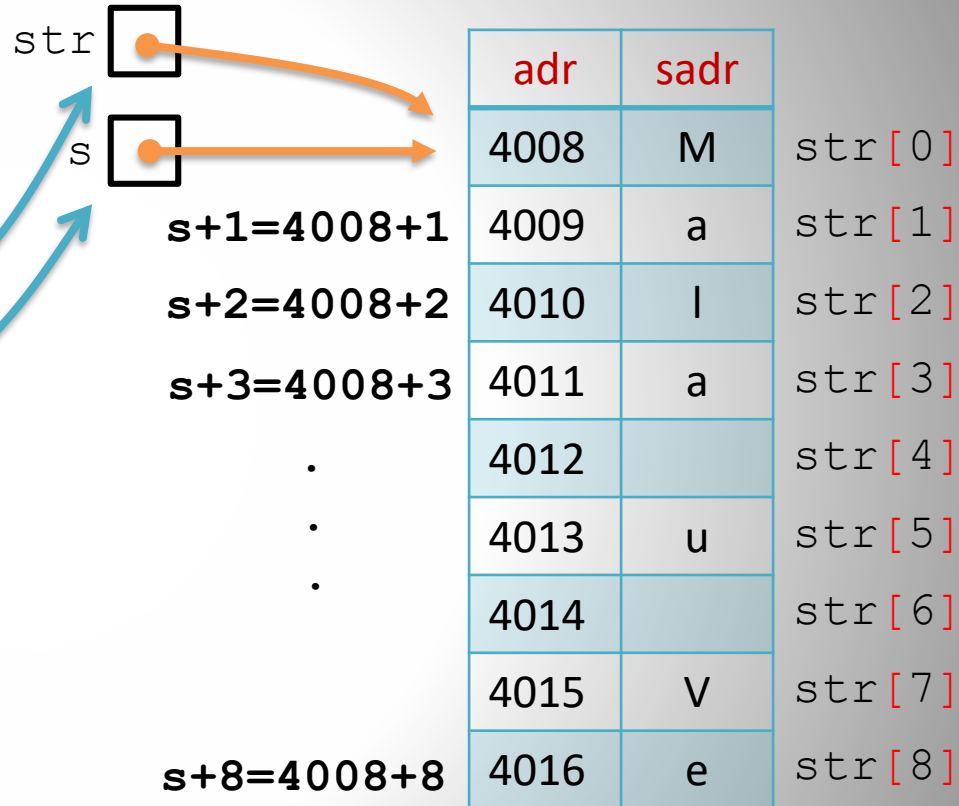
int main()
{
    char str[100];
    char* s;

    printf("Uneti string: ");
    gets(str);

    s=str;

    while(*s) {
        if(isupper(*s)) {
            *s=tolower(*s);
        }
        else {
            if(islower(*s)) {
                *s=toupper(*s);
            }
        }
        s++;
    }

    printf("String: %s", str);
    return 0;
}
```



```
Uneti string: Mala u Velika Slova
String: mALA U vELIKA sLOVA
Process returned 0 (0x0) execution time : 11.003 s
Press any key to continue.
```

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

int main()
{
    char str[100];
    char* s;

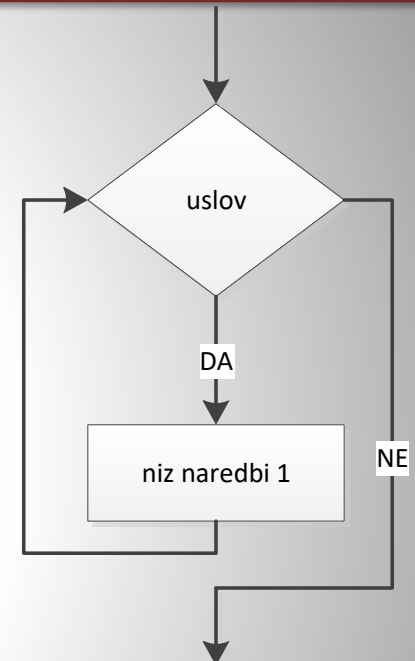
    printf("Uneti string: ");
    gets(str);

    s=str;

    while(*s) {
        if(isupper(*s)) {
            *s=tolower(*s);
        }
        else {
            if(islower(*s)) {
                *s=toupper(*s);
            }
        }
        s++;
    }

    printf("String: %s", str);
    return 0;
}
```

**while (izraz)
naredba petlje;**



U while-petlji se prvo izračunava vrednost izraza u zagradi. Ukoliko je **izraz** logički tačan, tj. ukoliko je njegova brojna vrednost **različita od nule**, izvršiće se naredba petlje. Nakon izvršenja ceo postupak će se ponoviti izračunavanjem izraz-a, sve dok on ne dobije logičku vrednost **lažan**.

0	1	2	3	4	5	6	7	8	...	19
P	o	z	s	r	a	v	\0	?	...	?

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

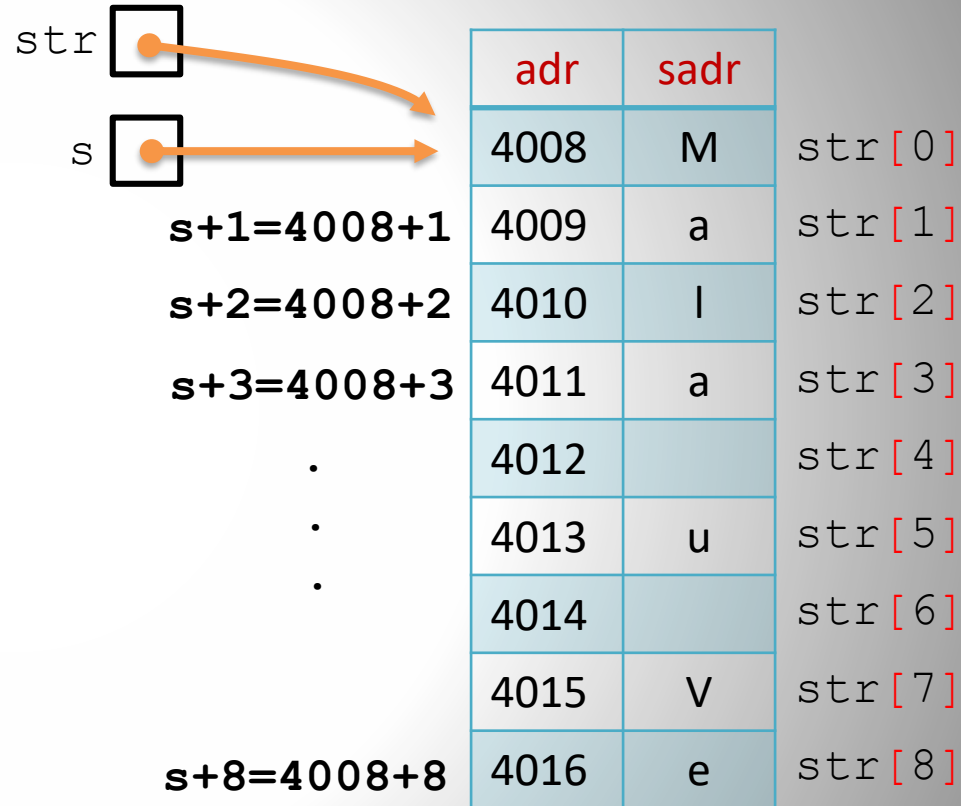
int main()
{
    char str[100];
    char* s;

    printf("Uneti string: ");
    gets(str);

    s=str;

    while(*s) {
        if(isupper(*s)) {
            *s=tolower(*s);
        }
        else {
            if(islower(*s)) {
                *s=toupper(*s);
            }
        }
        s++;
    }

    printf("String: %s", str);
    return 0;
}
```



```
Uneti string: Mala u Velika Slova
String: mALA U vELIKA sLOVA
Process returned 0 (0x0) execution time : 11.003 s
Press any key to continue.
```

- Kao što funkciji prosleđujemo parametre, nekada želimo da odmah prosledimo neke parametre programu koji smo pokrenuli.
- Primer je program COPY koji možemo pokrenuti iz komandne linije.
- Kada ga pokrećemo, pišemo razmak, ime fajla koji treba kopirati, pa razmak i ime novog fajla u koji se sadržaj kopira.

```
e:\>copy br_red.exe novi.exe  
1 file(s) copied.
```

- Da bi smo mogli da pišemo slične programe, moramo znati kako da preuzmemo te parametre (argumente) u naš program.
 - Do sada smo navikli da pišemo funkciju `main` kao:
- ```
int main () { ... }
```
- Međutim, `main` funkcija ima i svoj drugi zapis koji nam omogućava čitanje parametara koji su prosleđeni programu iz komandne linije:

```
int main (int argc, char **argv)
```

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

```
int main(int argc, char** argv)
```

```
{
```

```
 int i;
```

```
 printf ("Broj argumenata komandne linije je %d\n", argc);
```

```
 printf ("Parametri komandne linije: \n");
```

```
 for (i=0; i<argc; i++) {
```

```
 printf ("%s\n", argv[i]);
```

```
 }
```

```
 return 0;
```

```
}
```

```
e:\>prg prvi drugi treci cetvrti
Broj argumenata komandne linije je 5
Parametri komandne linije:
prg
prvi
drugi
treci
cetvrti
```