



Osnovi programiranja

Vežbe 10

ŠKOLSKA 2025/2026 GODINA

PRIRODNO-MATEMATIČKI FAKULTET

INSTITUT ZA MATEMATIKU I INFORMATIKU

- Funkcije za rad sa stringovima: **<string.h>**

"A String"

A		S	t	r	i	n	g	\0
---	--	---	---	---	---	---	---	----

- **"Hello world"** – primer stringa
- **" "** – prazan string
- Stringovi su nizovi karaktera koji se završavaju znakom **'\0'**.
- **'a' nije isto što i "a"**
- **'a'** – ceo broj koji predstavlja numeričku vrednost slova a u skupu znakova koje računar koristi
- **"a"** – niz znakova koji sadrži znak **'a'** i **'\0'**
- Znakovni nizovi (stringovi) se mogu inicijalizovati pomoću navodnika
- **char s[] = "A string";**
- što je skraćeni zapis od **char s[] = { 'A', ' ', 's', 't', 'r', 'i', 'n', 'g', '\0' };**
- Kreiranje string promenljive: **char s[20];**
- Stringovi - učitavanje: **scanf("%s", s);**
- Stringovi - štampanje : **printf("%s", s);**

Napisati funkciju **duzina_stringa** koja za prosleđeni string računa njegovu dužinu?

STRLEN

```
#include <stdio.h>
#include <string.h>
/* Izracunava duzinu stringa */
int duzina_stringa(char s[])
{
    int i;
    for (i = 0; s[i]!='\0'; i++) ;
    return i;
}
main()
{
    char string[] = "Primer stringa za testiranje";
    int moja_duzina = duzina_stringa(string);
    int ugrađena_duzina = strlen(string);
    printf("String: \"%s\"\n", string);
    printf("Duzina izracunata funkcijom duzina_stringa: %d\n", moja_duzina);
    printf("Dužina izračunata ugrađenom funkcijom strlen: %d\n", ugrađena_duzina);
}
```

Napisati funkciju koja kopira jedan string u drugi, pretpostaviti da ima dovoljno prostora?

```
#include <stdio.h>
#include <string.h>
/* Kopira string src u string dest.
Pretpostavlja da u dest ima dovoljno
prostora.
*/
void kopiraj_string(char dest[], char src[])
{
    int i;
    /* Kopira karakter po karakter, sve dok
    nije iskopiran karakter '\0' */
    for (i = 0; src[i] != '\0'; i++)
        dest[i] = src[i];
    dest[i] = '\0';
}
```

```
main()
{
    char original[] = "Primer za kopiranje";
    char kopija1[100];
    char kopija2[100];

    kopiraj_string(kopija1, original);

    strcpy(kopija2, original);

    printf("Original: \"%s\"\n", original);
    printf("Kopija 1: \"%s\"\n", kopija1);
    printf("Kopija 2: \"%s\"\n", kopija2);
}
```

Napisati funkciju **kopiraj_deo** koja jedan string kopira u drugi od date pozicije za dat broj karaktera?

```
#include <stdio.h>
#include <string.h>
void kopiraj_deo(char dest[], char src[], int poz, int br) {
    int i;
    for (i = 0; i < br; i++)
        dest[i] = src[poz + i];
    dest[i] = '\\0'; // Dodaj kraj stringa
}

int main() {
    char original[] = "Ovo je testiranje funkcije.";
    char kopirani_deo[100];
    int pocetna_pozicija = 7; // Pocinje od "t"
    int broj_karaktera = 10;

    kopiraj_deo(kopirani_deo, original, pocetna_pozicija, broj_karaktera);

    printf("Originalni string: \"%s\\n\"", original);
    printf("Kopirani deo: \"%s\\n\"", kopirani_deo);
}
```

Napisati funkciju **nadovezi_stringove** koja nadovezuje dva stringa?

```
#include <stdio.h>
#include <string.h>
/* Nadovezuje string t na kraj stringa s. Pretpostavlja da u s ima dovoljno
prostora. */
void nadovezi_stringove(char s[], char t[]) {
    int i, j;
    /* Pronalazimo kraj stringa s */
    for (i = 0; s[i] != '\0'; i++);
    /* Kopiranje karaktera iz t u s */
    for (j = 0; (s[i] = t[j]) != '\0'; j++, i++);
}
main() {
    char prvi[100] = "Ovo je prvi string.";
    char drugi[] = " Ovo je drugi string.";

    nadovezi_stringove(prvi, drugi);

    printf("Rezultat nadovezivanja: \"%s\"\n", prvi);

    char treci[100] = "Ovo je prvi string.";
    strcat(treci, drugi);
    printf("Rezultat strcat funkcije: \"%s\"\n", treci);
}
```

Ova funkcija poredi dva stringa i vraća:

1. 0 ako su jednaki
2. negativan broj ako je prvi string “manji od” drugog stringa leksikografski
3. pozitivan broj u suprotnom

```
#include <stdio.h>
#include <string.h>
main() {
    char string1[] = "abc";
    char string2[] = "abd";

    int rezultat = strcmp(string1, string2);

    if (rezultat == 0)
        printf("Stringovi \"%s\" i \"%s\" su jednaki.\n", string1, string2);
    else if (rezultat < 0)
        printf("String \"%s\" je leksikografski manji od stringa \"%s\".\n", string1, string2);
    else
        printf("String \"%s\" je leksikografski veći od stringa \"%s\".\n", string1, string2);
}
```

Napisati funkciju **string_char** koja za prosleđeni string i karakter ispituje da li se karakter nalazi u stringu i vraća poziciju prvog pojavljivanja?

```
#include <stdio.h>
/* Pronalazi prvu poziciju karaktera c u stringu s,
   odnosno -1 ukoliko s ne sadrži c. */
int string_char(char s[], char c) {
    int i;
    for (i = 0; s[i] != '\0'; i++)
        if (s[i] == c)
            return i; // Vraća indeks prvog pojavljivanja
    /* Nije nađeno */
    return -1;
}

main() {
    char string[] = "Ovo je primer stringa.";
    char karakter = 'r', karakter_nepostoji = 'x';

    printf("Karakter '%c': pozicija %d\n", karakter, string_char(string, karakter));
    printf("Karakter '%c': pozicija %d\n", karakter_nepostoji, string_char(string,
    karakter_nepostoji));
}
```

Napisati funkciju **string_string** koja proverava da li jedan string sadrži drugi string, ukoliko sadrži odrediti poziciju podstringa.

```
/* Proverava da li string str sadrzi string sub.  
   Vraca poziciju na kojoj sub pocinje, odnosno - 1 ukoliko ga nema */  
int string_string(char str[], char sub[])  
{  
    int i, j;  
    /* Proveravamo da li sub pocinje na svakoj poziciji i */  
    for (i = 0; str[i]; i++)  
        /* Poredimo sub sa str pocevsi od poziciji i sve dok ne naidjemo na razliku */  
        for (j = 0; str[i+j] == sub[j]; j++)  
            /* Nismo naisli na razliku a ispitali smo sve karaktere niske sub */  
            if (sub[j+1] == '\\0')  
                return i;  
    /* Nije nadjeno */  
    return -1;  
}
```

Napisati funkciju **delete_deo** koja za prosleđeni string briše deo stringa od date pozicije i broja karaktera?

```
#include <stdio.h>
/* Brise deo stringa od pozicije `poz` u dužini `br`. */
void delete_deo(char str[], int poz, int br) {
    int i;
    for (i = 0; str[i + poz + br] != '\0'; i++)
        str[poz + i] = str[poz + br + i];
    str[poz + i] = '\0';
}
main() {
    char string[] = "Ovo je primer stringa.";
    int pozicija = 7; // Pocetak za brisanje (karakter 'p')
    int duzina = 7;    // Duzina dela za brisanje ("primer")

    printf("Originalni string: \"%s\"\n", string);

    // Brisanje dela stringa
    delete_deo(string, pozicija, duzina);

    printf("Nakon brisanja: \"%s\"\n", string);
}
```

Napisati program koji uneti string u format

Prezime*Broj*Ime

prevodi u format

Ime Prezime Broj .

primer:

Peric*52*Pera

prevodi u

Pera Peric 52

```
main()
{
    char s[50],p[20],b[20]; int n;
    scanf("%s",s);
    n = string_char(s,'*');
    kopiraj_deo(p,s,0,n);
    delete_deo(s,0,n+1);
    n = string_char(s,'*');
    kopiraj_deo(b,s,0,n);
    delete_deo(s,0,n+1);
    strcat(s," ");
    strcat(s,p);
    strcat(s," ");
    strcat(s,b);
    printf("%s",s);
}
```

Napisati funkciju koja za unetu reč određuje da li je palindrom ili ne. Za niz unetih reči štampati palindrome među njima.



```
#include <stdio.h>
#include <string.h>
int palindrom(char rec[])
{
    int i,j,duzinaReci = strlen(rec);
    int jestePalindrom=0;
    i = 0;
    j = duzinaReci-1;
    while((rec[i]==rec[j]) && (i<j))
    {
        i++;
        j--;
    }
    if (i<j) return 0;
    else return 1;
}
```

```
main()
{
    char nizReci[100][100];
    int n,i;
    printf("Koliko ima reci? \n");
    scanf("%d",&n);
    for (i=0;i<n;i++)
        scanf("%s",&nizReci[i]);
    for (i=0;i<n;i++)
        if (palindrom(nizReci[i]))
            printf("%s\n",nizReci[i]);
}
```

Hvala na pažnji!