

# Programiranje I

*Beleške sa vežbi*

Smer *Informatika*  
Matematički fakultet, Beograd

Sana Stojanović

December 18, 2007

## Sadržaj

1 Makroi	3
2 Karakterske niske	4
3 Prenos nizova u funkciju	4
4 Funkcije za rad sa niskama	5

# 1 Makroi

## 1. Demonstracija pretprocesorske direktive `#define`

```
#include<stdio.h>

/* Racuna sumu dva broja */
#define zbir(a,b) ((a)+(b))

/* Racuna kvadrat broja - netacna verzija */
#define kvadrat_p(a) a*a

/* Racuna kvadrat broja */
#define kvadrat(a) ((a)*(a))

/* Racuna minimum tri broja */
#define min(a, b, c) (a)<(b) ? ((a)<(c)) ? (a) : (c) : ((b)<(c) ? (b) : (c))

main()
{
    printf("zbir(3,5) = %d\n", zbir(3,5));
    printf("kvadrat_p(5) = %d\n", kvadrat_p(5));
    printf("kvadrat_p(3+2) = %d\n", kvadrat_p(3+2));
    printf("kvadrat(3+2) = %d\n", kvadrat(3+2));
    printf("min(1,2,3) = %d\n", min(1,2,3));
    printf("min(1,3,2) = %d\n", min(1,3,2));
    printf("min(2,1,3) = %d\n", min(2,1,3));
    printf("min(2,3,1) = %d\n", min(2,3,1));
    printf("min(3,1,2) = %d\n", min(3,1,2));
    printf("min(3,2,1) = %d\n", min(3,2,1));
}
```

Izlaz iz programa:

```
zbir(3,5) = 8
kvadrat_p(5) = 25
kvadrat_p(3+2) = 11
kvadrat(3+2) = 25
min(1,2,3) = 1
min(1,3,2) = 1
min(2,1,3) = 1
min(2,3,1) = 1
min(3,1,2) = 1
min(3,2,1) = 1
```

## 2 Karakterske niske

```
1. /* Program uvodi niske karaktera terminisane nulom */

#include <stdio.h>

main()
{
    /* Poslednji bajt niske karaktera se postavlja na '\0' tj. 0 */
    char s[] = {'a', 'b', 'c', '\0' };

    /* Kraci nacin da se postigne prethodno */
    char t[] = "abc";

    /* Ispis niske s karakter po karakter*/
    int i;
    for (i = 0; s[i] != '\0'; i++)
        putchar(s[i]);
    putchar('\n');

    /* Ispis niske s koristeći funkciju printf */
    printf("%s\n", s);

    /* Ispis niske t karakter po karakter*/
    for (i = 0; t[i] != '\0'; i++)
        putchar(t[i]);
    putchar('\n');

    /* Ispis niske t koristeći funkciju printf */
    printf("%s\n", t);

}
```

## 3 Prenos nizova u funkciju

```
1. /* Funkcija za ispis niza brojeva - demonstrira prenos nizova
    brojeva u funkciju */
#include <stdio.h>

/* Nizovi se prenose tako sto se prenese adresa njihovog pocetka.
    Uglaste zagrade ostaju prazne!
    Nizove je neophodno prenositi zajedno sa dimenzijom niza
    (osim niski karaktera)
*/
```

```

void print_array(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ",a[i]);
    putchar('\n');

    /* Obratite paznju na ovo : */
    printf("sizeof(a) - u okviru fje : %d\n", sizeof(a));
}

main()
{
    int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
    int n;

    //racunamo dimenziju niza
    n = sizeof(a)/sizeof(int);

    printf("sizeof(a) - u okviru main : %d\n", sizeof(a));
    //stampamo niz pozivom funkcije
    print_array(a, n);
}

```

2. /\* Funkcija za ispis niske karaktera - demonstrira prenos niske karaktera u funkciju\*/

```

#include <stdio.h>

/* Uz nisku karaktera nije potrebno prenositi dimenziju ukoliko se postuje
dogovor da se svaka niska završava karakterom '\0'.*/
void print_string(char s[])
{
    int i;
    for (i = 0; s[i]; i++)
        putchar(s[i]);
}

main()
{
    print_string("Zdravo\n");
}

```

## 4 Funkcije za rad sa niskama

1. Funkcija koja računa dužinu niske.

```

#include <stdio.h>

int string_length(char s[])
{
    int i;
    for (i = 0; s[i]; i++)
        ;

    return i;
}

main()
{
    char s[] = "abc";

    printf("Duzina niske %s je %d\n", s, string_length(s));
}

```

2. Napisati funkciju koja obrće datu nisku karaktera (bez korišćenja dodatnih nizova).

```

/* string_reverse - obrce nisku karaktera */

#include <stdio.h>

/* Ova funkcija racuna duzinu date niske karaktera.
   Umesto nje, moguće je koristiti standardnu funkciju strlen .
*/
int string_length(char s[])
{
    int i;
    for (i = 0; s[i]; i++)
        ;

    return i;
}

/* Funkcija obrce nisku karaktera */
void string_reverse(char s[])
{
    int i, j;
    for (i = 0, j = string_length(s)-1; i<j; i++, j--)
    {
        int tmp = s[i];
        s[i] = s[j];
        s[j] = tmp;
    }
}

```

```

    }
}

main()
{
    char s[] = "Zdravo svima";
    string_reverse(s);
    printf("%s\n", s);
}

```

### 3. Ilustracija funkcija za rad sa niskama.

```

/* strlen, strcpy, strcat, strcmp, strchr, strstr -
   manipulacija niskama karaktera */
/* Vezbe radi, implementirane su funkcije biblioteke string.h */
#include <stdio.h>

/* Izracunava duzinu stringa */
int string_length(char s[])
{
    int i;
    for (i = 0; s[i]; i++)
        ;
    return i;
}

/* Kopira string src u string dest.
   Pretpostavlja da u dest ima dovoljno prostora. */
void string_copy(char dest[], char src[])
{
    /*Kopira karakter po karakter,sve dok nije iskopiran karakter '\0' */
    int i;
    for (i = 0; (dest[i]=src[i]) != '\0'; i++)
        ;

    /* Uslov != '\0' se, naravno, moze izostaviti :

    for (i = 0; dest[i]=src[i]; i++)
        ;
    */
}

/* Nadovezuje string t na kraj stringa s.

```

```

    Pretpostavlja da u s ima dovoljno prostora. */
void string_concatenate(char s[], char t[])
{
    int i, j;
    /* Pronalazimo kraj stringa s */
    for (i = 0; s[i]; i++)
        ;

    /* Vrsi se kopiranje, slicno funkciji string_copy */
    for (j = 0; s[i] = t[j]; j++, i++)
        ;
}

/* Vrsi leksikografsko poredjenje dva stringa.
   Vraca :
       0 - ukoliko su stringovi jednaki
       <0 - ukoliko je s leksikografski ispred t
       >0 - ukoliko je s leksikografski iza t
*/
int string_compare(char s[], char t[])
{
    /* Petlja tece sve dok ne naidjemo na prvi razliciti karakter */
    int i;
    for (i = 0; s[i]==t[i]; i++)
        if (s[i] == '\0') /* Naisli smo na kraj oba stringa,
                           a nismo nasli razliku */
            return 0;

    /* s[i] i t[i] su prvi karakteri u kojima se niske razlikuju.
       Na osnovu njihovog odnosa, odredjuje se odnos stringova */
    return s[i] - t[i];
}

/* Pronalazi prvu poziciju karaktera c u stringu s, odnosno -1
   ukoliko s ne sadrzi c */
int string_char(char s[], char c)
{
    int i;
    for (i = 0; s[i]; i++)
        if (s[i] == c)
            return i;
    /* nikako
       else
            return -1;
    */
    /* Nije nadjeno */
}

```



```

        return -1;
    }

    /* Pronalazi poslednju poziciju karaktera c u stringu s, odnosno -1
       ukoliko s ne sadrzi c */
    int string_last_char(char s[], char c)
    {
        /* Pronalazimo kraj stringa s */
        int i;
        for (i = 0; s[i]; i++)
            ;

        /* Krecemo od kraja i trazimo c unazad */
        for (i--; i>=0; i--)
            if (s[i] == c)
                return i;

        /* Nije nadjeno */
        return -1;

        /*
        Koristeci string_length :

        for (i = string_length(s) - 1; i>0; i--)
            if (s[i] == c)
                return i;

        return -1;
        */
    }

    /* Efikasnija verzija funkcije za pronalazenje poslednje pozicije */
    int string_last_char_ef(char s[], char c)
    {
        int i;
        int p = -1;

        //u jednom prolazu pretrazujemo string i pamtimo svako pojavljivanje
        //karaktera
        for(i=0; s[i]; i++)
            //ako nadjemo karakter, pamtimo poziciju
            if (s[i] == c)
                p = i;

        return p;
    }
}

```

```
main()
{
    char s[100];
    char t[] = "Zdravo";
    char u[] = " svima";

    //poziv funkcije koja kopira nisku t u nisku s
    string_copy(s, t);
    printf("%s\n", s);

    //poziv funkcije koja nadovezuje nisku u na nisku s
    string_concatenate(s, u);
    printf("%s\n", s);

    //pronalazenje prve i poslednje pozicije karaktera u nisci
    printf("%d\n",string_char("racunari", 'n'));
    printf("%d\n",string_last_char("racunari", 'a'));
}
```