

Programiranje 1
Beleške sa vežbi
Školska 2008/2009 godina

Matematički fakultet, Beograd

Jelena Graovac (Tomašević)

December 24, 2008

Sadržaj

1 Programski jezik C	5
1.1 Prenos niza u f-ju	5
1.2 Funkcije za rad sa niskama (stringovima)	6

1

Programski jezik C

1

1.1 Prenos niza u f-ju

Nizovi se prenose u funkciju tako što se prenese adresa njihovog početka. Iz tog razloga oni se MENJAJU u okviru funkcije.

Napomena: Sve nizove osim niski karaktera (stringova) neophodno je prenositi zajedno sa dimenzijom niza.

Primer 1 *Funkcija za ispis niza brojeva - demonstracija prenosa niza brojeva u funkciju.*

```
#include <stdio.h>

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

    /* Obratite paznju na ovo : */
    /* Ispisace se broj bajtova koje zauzima promenljiva
    tipa pokazivac na int */
    printf("sizeof(a) - u okviru fje : %d\n", sizeof(a));
}

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

    /* Ispisace se broj bajtova koje zauzima niz a */
    printf("sizeof(a) - u okviru main : %d\n", sizeof(a));
    stampaj_niz(a, sizeof(a)/sizeof(int));
}
```

¹Zasnovano na primerima sa sajta <http://www.matf.bg.ac.yu/~filip>.

```
Izlaz:
sizeof(a) - u okviru main : 36
1 2 3 4 5 6 7 8 9
sizeof(a) - u okviru fje : 4
```

Primer 2 *Skalarni proizvod dva niza brojeva*

```
#include <stdio.h>

long mnozi(int x[],int y[],int n);

main()
{
    int a[]={1,2,3,4,5,6}, b[]={8,7,6,5,4,3};
    printf("Skalarno a*b= %ld\n",mnozi(a,b,6));
}
long mnozi(int x[ ],int y[ ],int n)
{
    int i;
    long suma=0;
    for(i=0;i<n;i++) suma=suma+x[i]*y[i];
    return suma;
}
```

```
Izlaz:
Skalarno a*b= 98
```

1.2 Funkcije za rad sa niskama (stringovima)

Niska karaktera ili string je niz karaktera koji se završava karakterom '\0'.

Napomena: Karakter '\0' ima ASCII vrednost 0 pa se može tumačiti kao logička vrednost "netačno".

Format za ispis niske pomoću funkcije printf je %s. Konstante tipa niska se navode između znakova navodnika. Na primer, možemo napisati:

```
char s[]="Primer stringa";
```

Pri tome je s niska od 15 karaktera:

```
{'P', 'r', 'i', 'm', 'e', 'r', ' ', 's', 't', 'r', 'i', 'n', 'g', 'a', '\0'}
```

Primer 3 *Kakva je razlika između 's' i "s"?*

's' je karakter "s" je string ili niz od dva karaktera 's' i '\0'.

Primer 4 *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 stampaj_nisku(char s[])
```

```
{
    int i;
    for (i = 0; s[i]; i++)
        /* s[i] ce biti netacno samo kada bude jednako sa '\0' ciji je ASCII kod 0. */
        /* Ovo je isto kao da smo napisali for(i = 0; s[i]!='\0'; i++) */
        putchar(s[i]);
}
```

```
main()
{
    stampaj_nisku("Zdravo\n");
}
Izlaz:
Zdravo
```

Primer 5 Funkcija za učitavanje reči sa ulaza u nisku karaktera.

```
#include <stdio.h>
#include <ctype.h> /* Potrebno je zbog funkcije isspace */

/* Funkcija ucitava rec sa standardnog ulaza i smesta je u niz karaktera s.
   Ovo uspeva zbog toga sto se po vrednosti prenosi adresa pocetka niza,
   a ne ceo niz */
void ucitaj_rec(char s[])
{
    int c, i = 0;

    /* Funkcija isspace ispituje da li je karakter praznina
       (blanko, tabulator ili prelazak u novi red).
       Ona je definisana u okviru ctype.h */
    while (!isspace(c=getchar()))
        s[i++] = c;
    s[i] = '\0';
}

main()
{
    /* Obavezno je alocirati memoriju za niz karaktera */
    char s[100];

    ucitaj_rec(s);
    /* Format za ispis stringa se zadaje kao %s */
    printf("%s\n", s);
}
```

Da li bi bio ispravan sledeći program:

```
#include <stdio.h>

/* Funkcija ucitava ceo broj sa standardnog ulaza i smesta je u promenljivu x. */
void ucitaj_broj(int x)
{
```

```

    scanf("%d", &x);
}

main()
{
    int x;
    ucitaj_broj(x);
    printf("%d\n", x);
}

```

Ako ovo nije ispravno, zašto nije?

Primer 6 *obrni_string* - obrće nisku karaktera.

```

#include <stdio.h>

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

    return i;
}

/* Funkcija obrće nisku karaktera */
void obrni_string(char s[])
{
    int i, j;
    for (i = 0, j = duzina_stringa(s)-1; i<j; i++, j--)
    {
        int pom = s[i];
        s[i] = s[j];
        s[j] = pom;
    }

    /* Napomena : razlikovati prethodnu petlju od dve ugnjezdjene petlje:
    for ( i = 0; ....)
        for ( j = duzina(s)-1; ...
    */
}

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

```


Izlaz:
amivs ovarDZ

Primer 7 *Uklanja beline, tabulatore ili znak za kraj reda sa kraja stringa*

```
int ukloni(char s[])
{
    int i;
    for (i = strlen(s)-1; i >= 0; i--)
        if (s[i] != ' ' && s[i] != '\t' && s[i] != '\n')
            break;
    s[i+1] = '\0';
    return i;
}
```

Continue se rede koristi, on prouzrokuje da se pređe na sledeću iteraciju u petlji.

Primer 8

```
for(i=0; i<n; i++)
{
    if (a[i]==0) continue; ... /* obradi pozitivne elemente nekako*/
}
```

Primer 9 *Izvršiti implementaciju funkcija strlen, strcpy, strcat, strcmp, strchr, strstr biblioteke string.h*

```
#include <stdio.h>

/* Izracunava duzinu stringa */
int duzina_stringa(char s[])
{
    int i;
    for (i = 0; s[i]; i++)
/* for (i = 0; s[i]!='\0'; i++) */
        ; /*Telo petlje je prazno*/
    return i;
}

/* 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];

    /* Ovo se moglo zapisati i kao:
       for (i = 0; (dest[i]=src[i]) != '\0'; i++)
           ;
    */
}
```

```
    */
    /* a ako se izostavi uslov != '\0', moze se zapisati kao:

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

/* 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]; i++)
        ;

    /* Vrsi se kopiranje, slicno funkciji kopiraj_string */
    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 uporedi_stringove(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_poslednji_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 duzina_stringa :

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

    return -1;
    */
}

/* 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;
}
```

```

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

    kopiraj_string(s, t);
    printf("%s\n", s);

    nadovezi_stringove(s, u);
    printf("%s\n", s);

    printf("%d\n", string_char("racunari", 'n'));
    printf("%d\n", string_poslednji_char("racunari", 'a'));

    printf("%d\n", string_string("racunari", "rac"));
    printf("%d\n", string_string("racunari", "ari"));
    printf("%d\n", string_string("racunari", "cun"));
    printf("%d\n", string_string("racunari", "cna"));
}

```

```

Izlaz:
Zdravo
Zdravo svima
4
5
0
5
2
-1

```

Primer 10 Funkcija koja uklanja znak *c* kad god se pojavi u stringu *s*.

```

#include <stdio.h>
void sazimanje(char s[], char c)
{
    int i,j;
    for(i=j=0; s[i]!='\0';i++)
        if(s[i]!=c) s[j++]=s[i];
    s[j]='\0';
}

main() {
    char niz[20];
    char c;

    printf("Unesi karakter\n\n");
    scanf("%c", &c);

    scanf("%s", &niz);
    sazimanje(niz, c);
    printf("%s\n", niz);
}

```

```
}

```

Izlaz:

```
Unesi karakter
i
Unesi string
primer
prmer
```

Zadaci za vežbu

Zadatak 1 Napisati funkciju koja vraća prvu poziciju u niski s_1 na kojoj se pojavljuje znak iz s_2 ili -1 ako s_1 ne sadrži ni jedan znak iz s_2 . Ako je s_1 pera a s_2 navip onda funkcija treba da vrati poziciju 0. Ako je s_1 zeleno a s_2 nana onda funkcija treba da vrati poziciju 4.

Zadatak 2 januar 2006. (II grupa)

1. Napisati funkciju **void brojanje(int a[], int brojac[], int N)** čiji su argumenti a i $brojac$ celobrojni nizovi dimenzije N . Vrednosti elemenata niza a su između 0 i $N - 1$. Funkcija izračunava elemente niza $brojac$ tako da je $brojac[i]$ jednak broju pojavljivanja broja i u nizu a .
2. Kažemo da je celobrojni niz a dimenzije N permutacija ako sadrži svako i : $0 \leq i < N$. Sastaviti funkciju **int DaLiJePermutacija(int a[], int N)** koja vraća 1 ako je niz a permutacija, a inače 0 . (Koristiti funkciju `brojanje`)

Zadatak 3 I kolokvijum, 18. januar 2006. (I grupa) Neka je dat niz X od N nenegativnih celih brojeva. Sastaviti funkciju koja će iz niza X izbacivati sva pojavljivanja broja 0 i popunjavati ta mesta u nizu tako što će se preostali elementi niza pomerati ka početku niza. Odrediti i novu dimenziju N niza X . Npr. ulaz: $N = 10$, $X = 0\ 22\ 11\ 2\ 0\ 17\ 33\ 4\ 0\ 999$ izlaz: $N = 7$, $X = 22\ 11\ 2\ 17\ 33\ 4\ 999$.

Zadaci za praktikum

Zadatak 4 Napisati funkciju koja izračunava broj velikih slova, malih slova i cifara u tekstu kao i program za njeno testiranje. Broj velikih i broj malih slova vratiti preko liste argumenata.

Zadatak 5 Napisati funkciju koja izračunava koliko se puta data reč pojavila u tekstu i program koji vrši njeno testiranje. Reč učitati sa ulaza.

Zadatak 6 Napisati funkciju koja iz teksta učitava prvu reč koja sledi iza date reči i program koji vrši njeno testiranje. Reč učitati sa ulaza.

Zadatak 7 Napisati funkciju koja izračunava koliko ima reči u tekstu koje počinju i završavaju se istim karakterom i program koji vrši njeno testiranje.

Zadatak 8 Napisati funkciju koja izračunava da li je dati string palindrom. Napisati drugu funkciju koja računa koliko ima reči u tekstu koje su palindromi i program koji vrši njeno testiranje.