Programiranje 1

Beleške sa vežbi

Školska 2006/2007 godina

Matematički fakultet, Beograd

Jelena Tomašević

December 25, 2006

Sadržaj

1 Programski jezik C **5**
1.1 Prenos niza u f-ju 5
1.1.1 Funkcije za rad sa stringovima 6

1

Programski jezik C

[1]

1.1 Prenos niza u f-ju

Primer 1 *Demonstrira prenos nizova u funkciju - preneti niz se moze menjati.*

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

/* 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 get_word(char s[])
{
    int c, i = 0;
    while (!isspace(c=getchar()))
        s[i++] = c;
    s[i] = '\0';
}

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

    get_word(s);
    printf("%s\n", s);
}
```

Primer 2 *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!
```

[1] Zasnovano na primerima sa sajtova http://www.matf.bg.ac.yu/~filip,http://www.matf.bg.ac.yu/~milena.

```
   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};

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

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

Primer 3 *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 br;
   long suma=0;
   for(br=0;br<n;br++) suma=suma+x[br]*y[br];
   return suma;
}

Izlaz:
Skalarno a*b= 98
```

1.1.1 Funkcije za rad sa stringovima

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 zavrsava karakterom '\0'.*/
void print_string(char s[])
{
    int i;
    for (i = 0; s[i]; i++)
        putchar(s[i]);
}


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

Primer 5 *string_reverse - obrće nisku karaktera.*

```
#include <stdio.h>

/* Zbog funkcije strlen */
#include <string.h>

/* Ova funkcija racuna duzinu date niske karaktera.
   Umesto nje, moguce 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;
    }

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

```
     */

}

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

Izlaz:
amivs ovardZ
```

Primer 6 *Uklanja beline, tabulatore ili znak za kraj reda sa kraja stringa*

```
/* trim:  remove trailing blanks, tabs, newlines */
int trim(char s[])
{
int n;
for (n = strlen(s)-1; n >= 0; n--)
    if (s[n] != ' ' && s[n] != '\t' && s[n] != '\n')
                break;
s[n+1] = '\0';
return n;
}
```

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

Primer 7

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

Primer 8 *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;
    */
}

/* 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";

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

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

    printf("%d\n",string_char("racunari", 'n'));
    printf("%d\n",string_last_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 9 *Funkcija koja uklanja znak c kad god se pojavi u stringu s.*

```
#include <stdio.h>
void squeeze(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);
    squeeze(niz, c);
    printf("%s\n", niz);

}

Izlaz:
Unesi karakter
i
Unesi string
primer
prmer
```

Primer 10 *Dopisivanje stringova*

```
#include <stdio.h>

/* strcat:  concatenate t to
end of s; s must be big enough */
void strcat(char s[], char t[])
{
    int i, j;

    i = j = 0;
    while (s[i] != '\0') /* nadji kraj od s */
        i++;
    while ((s[i++] = t[j++]) != '\0') /* copy t */
        ;
}

main()
{
char spojeni[100]="Ovo je prvi";
char nastavak[100]=",a ovo nastavak";

printf("%s\n",spojeni);
printf("%s\n",nastavak);

strcat(spojeni,nastavak);

printf("%s\n",spojeni);
}

Izlaz:
Ovo je prvi
,a ovo nastavak
```

```
Ovo je prvi,a ovo nastavak
```

Zadaci za vežbu

Zadatak 1 *Napisati program koji u ucitanoj niski karaktera sa ulaza prebrojava pojavu cifara.*
Ilustracija redirekcije standardnog ulaza i izlaza :
pokrenuti program sa :
./a.out ¡zadatak.c ./a.out ¿tekst.txt ./a.out ¡zadatak.c ¿kopija.c

Zadatak 2 *Napisati funkciju koja vraća prvu poziciju u niski s1 na kojoj se pojavljuje znak iz s2 ili -1 ako s1 ne sadrži ni jedan znak iz s2. Ako je s1 pera a s2 navip onda funkcija treba da vrati poziciju 0. Ako je s1 zeleno a s2 nana onda funkcija treba da vrati poziciju 4.*

Zadatak 3 ***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<=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 4 ***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.`

Zadatak 5 ***I kolokvijum, februar 2005.***

1. *Napisati funkciju koja ispituje da li dve niske (koje se prenose kao parametri funkcije) su anagrami. Anagrami su niske koje se sastoje od istih karaktera. Npr. vetar, trave, verat su anagrami.*

2. *Napisati program koji testira funkciju iz prvog dela.*

Zadatak 6 ***I kolokvijum, februar 2005.*** *Napisati program koji učitava sa standardnog ulaza dve niske sa ne više od 80 karaktera u svakoj i prirodan broj* `k` *i ispisuje na standardni izlaz poruku da li se prva niska dobila cikličnim pomeranjem druge niske za* `k` *mesta. Na primer za* `k=3`*, niska CDEAB"" se dobila cikličnim pomeranjem niske "ABCDE"*

Zadatak 7 ***Septembar, 2005.*** *Napisati funkciju* `int poredi(char* p, char* d)` *koja vraća* `-1` *ukoliko je* `p<d`*,* `0` *ukoliko je* `p == d` *a 1 ako je* `p>d`*, pri čemu su* `p` *i* `d` *dva velika cela neoznačena broja zadata nizom(niskom) svojih cifara.*

Zadatak 8 ***Septembar, 2005.*** *Sa standardnog ulaza se učitava niz od* `n (n<100)` *tačaka u ravni takvih da nikoje tri tačke nisu kolinearne. Tačke se zadaju parom svojih koordinata (celi brojevi). Ispitati da li taj niz tačaka odredjuje konveksni mnogougao i rezultat ispisati na standardni izlaz.*

Zadatak 9 ***Jun, 2004.*** *Napisati funkciju koja koja kao argumente prihvata dve niske i proverava da li se prva od zadatih niski može dobiti cikličnim pomeranjem karaktera druge niske.*