Osnovi programiranja

Beleške sa vežbi

Smer *Računarstvo i informatika*
Matematički fakultet, Beograd

Jelena Tomašević

December 25, 2005

Sadržaj

1

[1]

1.1 Prenos parametara po vrednosti

Primer 1 *Demonstracija prenosa parametara po vrednosti - preneti parametri se ne mogu menjati*

```
#include <stdio.h>
void f(int x)
{
x++;
}

main()
{
int x=3;
f(x);
printf("%d\n", x);
}
Izlaz:
3
```

1.2 Prenos niza u f-ju

Primer 2 *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';
}
```

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

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

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

Primer 3 *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};

    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 4 *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
```

Primer 5 *Program vrši linearnu pretragu niza.*

```
#include <stdio.h>

/* Funkcija proverava da li se dati element x nalazi
   u datom nizu celih brojeva.
   Funkcija vraca poziciju u nizu na kojoj je x pronadjen
   odnosno -1 ukoliko elementa nema.
*/
int linearna_pretraga(int niz[], int br_elem, int x)
{
    int i;
    for (i = 0; i<br_elem; i++)
        if (niz[i] == x)
            return i;
    return -1;
}

main()
{
    int a[] = {4, 3, 2, 6, 7, 9, 11};
    int br_elem = sizeof(a)/sizeof(int);
    int x;
    int i;
    printf("Unesi broj koji trazimo : ");
    scanf("%d",&x);

    i = linearna_pretraga(a, br_elem, x);
    if (i == -1)
        printf("Element %d nije nadjen\n",x);
    else
        printf("Element %d je nadjen na poziciji %d\n",x, i);
}
```

1.2.1 Funkcije za rad sa stringovima

Primer 6 *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 7 *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 8 *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 9

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

Primer 10 *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 11 *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 12 *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.*

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 *Broj je* **Nivenov** *ako je deljiv sumom svojih cifara.*

1. *Napsati funkciju int sumaCifara(char a[]) koja računa sumu cifara broja* ***a****. Na primer, za "121" funkcija treba da vrati 4.*
2. *Napisati funkciju int Nivenov(char a[]) koja proverava da li je broj* ***a*** *Nivenov i vraća 1 ako jeste a 0 ako nije.*
3. *Napisati program koji sa ulaza učitava broj (kao nisku karaktera) i ispisuje na izlaz informaciju o tome da li je broj Nivenov.*

Zadatak 4 *Napisati program koji proverava da li je uneta niska palindrom. Na primer, palindrom je "anavolimilovana".*