

Osnovi programiranja

Beleške sa vežbi

Smer *Računarstvo i informatika*
Matematički fakultet, Beograd

Jelena Tomašević

December 25, 2005

Sadržaj

1		5
1.1	Funkcije za rad sa stringovima	5
1.1.1	Linearna i binarna pretraga niza	8
1.2	Makroi	8
1.3	Pokazivači - osnovni pojmovi	12

1

1

1.1 Funkcije za rad sa stringovima

Primer 1 Program pronalazi najdužu liniju sa ulaza.

```
#include <stdio.h>
#define MAX_DUZINA_LINIJE 80

int ucitaj_liniju(char linija[])
{
    int c;
    int pozicija = 0;
    while ((c=getchar())!=EOF && c != '\n' &&
           pozicija<MAX_DUZINA_LINIJE-1)
        linija[pozicija++]=c;

    if (c=='\n')
        linija[pozicija++] = c;

    linija[pozicija]=0;

    return pozicija;
}

/*
int duzina_linije(char linija[])
{
    int d;
    for (d=0; linija[d]!=0; d++)
        ;

    d=0;
    while (linija[d] != 0)
        d++;

    return d;
}
```

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

```
}
*/

void kopiraj_liniju(char u[], char iz[])
{
    int pozicija;
    pozicija = 0;
    do
    {
        u[pozicija] = iz[pozicija];
        pozicija++;
    } while (u[pozicija-1] != 0);

    /*
    do
    {
        u[pozicija] = iz[pozicija];
    } while (u[pozicija++] != 0);
    */

    /*
    pozicija = 0;
    while((u[pozicija] = iz[pozicija]) != 0)
        pozicija++;
    */

    /* while (a != 0) <==> while (a) */

    /* for (pozicija = 0; (u[pozicija] = iz[pozicija]) != 0; pozicija++)
        ;
    */

    /* for (pozicija = 0; u[pozicija] = iz[pozicija]; pozicija++)
        ;
    */

}

main()
{
    char linija[MAX_DUZINA_LINIJE];
    char najduza_linija[MAX_DUZINA_LINIJE];

    int duzina_linije = 0;
    int duzina_najduze_linije = 0;

    najduza_linija[0] = 0;

    while( (duzina_linije = ucitaj_liniju(linija)) != 0)
    {
```

```
        if (duzina_linije>duzina_najduze_linije)
        {
            kopiraj_liniju(najduza_linija, linija);
            duzina_najduze_linije = duzina_linije;
        }
    }

    printf("Najduza linija je : %s\n",najduza_linija);
}
```

Primer 2 Program pronalazi najduzu liniju sa ulaza(II verzija).

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

#define MAX_DUZINA_LINIJE 80

int učitaj_liniju(char linija[])
{
    int c;
    int pozicija = 0;
    while ((c=getchar())!=EOF &&
           c != '\n' &&
           pozicija<MAX_DUZINA_LINIJE-1)
        linija[pozicija++]=c;

    if (c=='\n')
        linija[pozicija++] = c;

    linija[pozicija]=0;

    return pozicija;
}

main()
{
    char linija[MAX_DUZINA_LINIJE];
    char najduza_linija[MAX_DUZINA_LINIJE];

    int duzina_linije = 0;
    int duzina_najduze_linije = 0;

    najduza_linija[0] = 0;

    while( (duzina_linije = učitaj_liniju(linija)) != 0)
    {
        if (duzina_linije>duzina_najduze_linije)
        {
            strcpy(najduza_linija, linija);
            duzina_najduze_linije = duzina_linije;
        }
    }
}
```

```

    }

    printf("Najduza linija je : %s\n", najduza_linija);
}

```

1.1.1 Linearna i binarna pretraga niza

Primer 3 *Linearno pretraživanje*

```

int linsearch(int x, int v[], int n)
{
    int i;
    for(i=0; i<n; i++)
        if(v[i]==x)
            return i;
    return -1; /* Nema poklapanja*/
}

```

Primer 4 *Binarno pretraživanje*

```

/* Napomena: Niz mora biti uredjen u rastucem redosledu. */
/* binsearch: find x in v[0] <= v[1]
<= ... <= v[n-1] */
int binsearch(int x, int v[], int n)
{
    int low, high, mid;

    low = 0;
    high = n - 1;
    while (low <= high) {
        mid = (low+high)/2;
        if (x < v[mid])
            high = mid - 1;
        else if (x > v[mid])
            low = mid + 1;
        else /* found match */
            return mid;
    }
    return -1; /* no match */
}

```

1.2 Makroi

Primer 5 *Šta je rezultat rada sledeceg programa :*

```

#include <stdio.h>
#define IN 1
#define OUT 0
main( )
{ int c, nw , state;
state=OUT;

```

```

nw=0;
while ( (c=getchar() ) != EOF)
{
if (c==' ' || c=='\n' || c=='\t') state=OUT;
else if (state==OUT) {
state=IN;
++nw;
}
}
printf("\n%d\n",nw);
}

```

Rezultat:

U sledecem novom redu se ispisuje broj reci teksta sa ulaza, pa se predje u novi red. Smatramo da rec je ma koji niz znakova koji ne sadri blanko, tab, new line.

Primer 6 *Demonstracija pretprocesorske direktive #define*

```

#include<stdio.h>
/* Racuna sumu dva broja */
#define sum(a,b) ((a)+(b))

/* Racuna kvadrat broja - pogresna verzija */
#define square_w(a) a*a

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

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

main()
{
printf("sum(3,5) = %d\n", sum(3,5));
printf("square_w(5) = %d\n", square_w(5));
printf("square_w(3+2) = %d\n", square_w(3+2));
printf("square(3+2) = %d\n", square(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:

```

sum(3,5) = 8
square_w(5) = 25
square_w(3+2) = 11
square(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
```

Primer 7 *Demonstracija pretprocesorske direktive #define*

```
#include <stdio.h>
#define KUBW(a) (a * a * a)
#define KUB(a)  ( (a) * (a) * (a))

main()

{

int b=1;

printf("KUB(%d) = %d\n", 2*b+4, KUBW(2*b+4) );
printf("KUB(%d) = %d\n", 2*b+4, KUB(2*b+4) );

}
```

Izlaz:

```
KUB(6) = 22
KUB(6) = 216
```

Primer 8 *Ilustracija beskonačne petlje:*

```
#define forever for(;;);
```

Moguće je definisati makroe sa argumentima tako da tekst zamene bude različit za različita pojavljivanja makroa.

Primer 9

```
#define max(A, B) ((A)>(B) ? (A) : (B))
```

na osnovu ovoga će linija

```
x=max(p+q, r+s)
```

biti zamenjena linijom

```
x=((p+q) > (r+s) ? (p+q) : (r+s));
```

Treba voditi računa o sporednim efektima. Sledeća linija koda prouzrokuje uvećanje vrednosti i i j za dva.

```
max(i++, j++)
```

Takođe treba voditi računa o zagradama. Sledeći makro prouzrokuje neočekivane rezultate za poziv square(a+1)

```
#define square(x) x*x
```

Primer 10

```
#include <stdio.h>
#define max1(x,y) (x>y?x:y)
#define max2(x,y) ((x)>(y)?(x):(y))
#define swapint(x,y) { int z; z=x; x=y; y=z; }
#define swap(t,x,y) { \
    t z; \
    z=x; \
    x=y; \
    y=z; }

main()
{

    int x=2,y=3;

    printf( "max1(x,y) = %d\n", max1(x,y) );
    /* max1(x,y) = 3 */

    /* Zamena makroom se ne vrši
    unutar niski pod navodnicima*/
    printf( "max1(x=5,y) = %d\n", max1(x,y) );
    /* max1(x=5,y) = 3 */

    printf( "max1(x++,y++) = %d\n", max1(x++,y++) );
    /* max1(x++,y++) = 4 */

    printf( "x = %d, y = %d\n", x, y );
    /* x = 3, y = 5 */

    swapint(x,y);

    printf( "x = %d, y = %d\n", x, y );
    /* x = 5, y = 3 */

    swap(int,x,y);
    printf( "x = %d, y = %d\n", x, y );
    /* x = 3, y = 5 */
}

Izlaz:
max1(x,y) = 3
max1(x=5,y) = 3
max1(x++,y++) = 4
x = 3, y = 5
x = 5, y = 3
x = 3, y = 5
```

1.3 Pokazivači - osnovni pojmovi

Primer 11 *Ilustracija rada sa pokazivačkim promenljivim.*

```
#include <stdio.h>
main()
{
    int x = 3;

    /* Adresu promenljive x zapamticemo u novoj promenljivoj.
       Nova promenljiva je tipa pokazivaca na int (int*) */
    int* px;

    printf("Adresa promenljive x je : %p\n", &x);
    printf("Vrednost promenljive x je : %d\n", x);

    px = &x;
    printf("Vrednost promenljive px je (tj. px) : %p\n", px);
    printf("Vrednost promenljive na koju ukazuje px (tj. *px) je : %d\n", *px);

    /* Menjamo vrednost promenljive na koju ukazuje px */
    *px = 6;
    printf("Vrednost promenljive na koju ukazuje px (tj. *px) je : %d\n", *px);

    /* Posto px sadrzi adresu promenljive x, ona ukazuje na x tako da je
       posredno promenjena i vrednost promenljive x */
    printf("Vrednost promenljive x je : %d\n", x);
}
```

Izlaz (u konkretnom slucaju):

```
Adresa promenljive x je : 0012FF88
Vrednost promenljive x je : 3
Vrednost promenljive px je (tj. px) : 0012FF88
Vrednost promenljive na koju ukazuje px (tj. *px) je : 3
Vrednost promenljive na koju ukazuje px (tj. *px) je : 6
Vrednost promenljive x je : 6
```

Primer 12 *swap : Demonstracija prenosa argumenata preko pokazivača.*

```
#include <stdio.h>

/* Pogresna verzija funkcije swap. Zbog prenosa po vrednosti, funkcija
   razmenjuje kopije promenljivih iz main-a, a ne samih promenljivih */
void swap_wrong(int x, int y)
{
    int tmp;

    printf("swap_wrong: ");
    printf("Funkcija menja vrednosti promenljivim na adresama : \n");
    printf("x : %p\n", &x);
```

```
printf("y : %p\n", &y);

tmp = x;
x = y;
y = tmp;
}

/* Resenje je prenos argumenata preko pokazivaca */
void swap(int* px, int* py)
{
    int tmp;

    printf("swap : Funkcija menja vrednosti promenljivim na adresama : \n");
    printf("px = %p\n", px);
    printf("py = %p\n", py);

    tmp = *px;
    *px = *py;
    *py = tmp;
}

main()
{
    int x = 3, y = 5;
    printf("Adresa promenljive x je %p\n", &x);
    printf("Vrednost promenljive x je %d\n", x);
    printf("Adresa promenljive y je %p\n", &y);
    printf("Vrednost promenljive y je %d\n", y);

    /* Pokušavamo zamenu koristeći pogrešnu verziju funkcije */
    swap_wrong(x, y);

    printf("Posle swap_wrong:\n");
    printf("Vrednost promenljive x je %d\n", x);
    printf("Vrednost promenljive y je %d\n", y);

    /* Vrsimo ispravnu zamenu. Funkciji swap saljemo adrese promenljivih
       x i y, a ne njihove vrednosti */
    swap(&x, &y);

    printf("Posle swap:\n");
    printf("Vrednost promenljive x je %d\n", x);
    printf("Vrednost promenljive y je %d\n", y);
}
```

Izlaz u konkretnom slucaju:

Adresa promenljive x je 0012FF88

Vrednost promenljive x je 3

Adresa promenljive y je 0012FF84

Vrednost promenljive y je 5

swap_wrong: Funkcija menja vrednosti promenljivim na adresama :

```

x : 0012FF78
y : 0012FF7C
Posle swap_wrong:
Vrednost promenljive x je 3
Vrednost promenljive y je 5
swap : Funkcija menja vrednosti promenljivim na adresama :
px = 0012FF88
py = 0012FF84
Posle swap:
Vrednost promenljive x je 5
Vrednost promenljive y je 3

```

Primer 13 *Demonstracija više povratnih vrednosti funkcije koristeći prenos preko pokazivača.*

```

/* Funkcija istovremeno vraća dve vrednosti - kolicnik i ostatak dva data broja.
   Ovo se postize tako sto se funkciji predaju vrednosti dva broja (x i y) koji se dele
   i adrese dve promenljive na koje ce se smestiti rezultati */
void div_and_mod(int x, int y, int* div, int* mod)
{
    printf("Kolicnik postavljam na adresu : %p\n", div);
    printf("Ostatak postavljam na adresu : %p\n", mod);
    *div = x / y;
    *mod = x % y;
}

main()
{
    int div, mod;
    printf("Adresa promenljive div je %p\n", &div);
    printf("Adresa promenljive mod je %p\n", &mod);

    /* Pozivamo funkciju tako sto joj saljemo vrednosti dva broja (5 i 2)
       i adrese promenljivih div i mod na koje ce se postaviti rezultati */
    div_and_mod(5, 2, &div, &mod);

    printf("Vrednost promenljive div je %d\n", div);
    printf("Vrednost promenljive mod je %d\n", mod);
}

```

Izlaz u konkretnom slucaju:
 Adresa promenljive div je 0012FF88
 Adresa promenljive mod je 0012FF84
 Kolicnik postavljam na adresu : 0012FF88
 Ostatak postavljam na adresu : 0012FF84
 Vrednost promenljive div je 2
 Vrednost promenljive mod je 1

Zadaci sa vežbi:

Zadatak 1 *Sastaviti C-direktivu koja određuje srednji po vrednosti od tri broja.*

Zadatak 2 *Sastaviti C-direktivu koja implementira operator implikacije.*

Zadatak 3 Učitava se linija po linija teksta. Odšampati svaku od tih linija tako da ima veliko slovo na početku rečenice i sva mala unutar rečenice(., ?, !).

Zadatak 4 Napisati funkciju koja izračunava celobrojni koren prirodnog broja i program koji tu funkciju poziva za vrednost unetu sa standardnog ulaza.

Zadatak 5 Napisati funkciju koja izračunava celobrojni količnik dva uneta cela broja i program koji tu funkciju poziva za vrednosti unete sa standardnog ulaza.

Zadatak 6 Napisati funkciju koja izračunava NZD dva uneta broja koristeći Euklidov algoritam i program koji tu funkciju poziva za vrednosti unete sa standardnog ulaza.

Zadatak 7 Napisati funkciju koja izračunava celobrojni koren prirodnog broja i program koji tu funkciju poziva za vrednost unetu sa standardnog ulaza.