

Programiranje 1

Beleške sa vežbi

Smer *Informatika*

Matematički fakultet, Beograd

Jelena Tomašević

December 11, 2005

Sadržaj

1		2
1.1	Formiranje HTML dokumenta	2
1.2	Strukture	3
1.3	Životni vek i oblast važenja promenljivih	7

Čas 1

1

1.1 Formiranje HTML dokumenta

Primer 1 *Prilikom pokretanja programa koristiti redirekciju:*

a.out >primer.html

kako bi se rezultat rada programa upisao u datoteku primer.html.

```
/*Ovaj program formira html dokument*/
#include <stdio.h>
main()
{
    printf("<html><head><title>Ova stranica
           je napravljena u c-u</title></head>");
    printf("<body><h3 align=center>
           Rezultat </h3></body></html>");
}
```

Primer 2 *Napisati program koji generiše html dokument sa engleskim alfabedom.*

```
#include <stdio.h>
main()
{
    int i;
    printf("<HTML><head><title>Engleski alfabet</title><head>\n");
    printf("<body><ul>");
    for(i=0;i<=25;i++)
        printf("<li> %c %c \n", 'A'+i, 'a'+i);
    printf("</ul></body></HTML>\n"); }
```

Primer 3 *Napisati program koji generise html dokument koji prikazuje tablicu mnozenja za brojeve od 1 do 10.*

```
#include<stdio.h>
main()
```

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

```

{
    int i,j;
    printf("<html><head><title>Mnozenje</title></head>");
    printf("<body><h3 align=center> Rezultat </h3>");
    printf("<table border=1>\n");

    /* Prva vrsta sadrzi brojeve od 1 do 10*/
    printf("<tr>");
    printf("<th></th>");
    for(i=1; i<=10; i++)
        printf("<th> %d </th>\n", i);
    printf("</tr>");

    for(i=1; i<=10; i++)
    {
        printf("<tr>");

        /* Na pocetku svake vrste stampamo broj
        odgovarajuće vrste*/
        printf("<th>%d</th>", i);

        for(j=1; j<=10; j++)
            printf("<td>%d\t</td>\n", i*j);

        printf("</tr>");
    }
    printf("</table>");
    printf("</body></html>");
    return 0;
}

```

1.2 Strukture

Primer 4 *Napisati program koji izračunava obim i površinu trougla i kvadrata.*

```

/* Program uvodi strukture - geometrijske figure */
#include <stdio.h>

/* Zbog funkcije sqrt. */
#include <math.h>
/* Upozorenje : pod linux-om je potrebno program prevoditi sa
gcc -lm primer.c
kada god se koristi <math.h>
*/

/* Tacke su predstavljene sa dve koordinate. Strukturom gradimo novi tip podataka. */
struct point
{
    int x;
    int y;
}

```

```
};

/* Izracunava duzinu duzi zadatu sa dve tacke */
float segment_length(struct point A, struct point B)
{
    int dx = A.x - B.x;
    int dy = A.y - B.y;
    return sqrt(dx*dx + dy*dy);
}

/* Izracunava povrsinu trougla Heronovim obrascem.
Argumenti funkcije su tri tacke koje predstavljaju temena trougla */
float Heron(struct point A, struct point B, struct point C)
{
    /* Duzine stranica */
    float a = segment_length(B, C);
    float b = segment_length(A, C);
    float c = segment_length(A, B);

    /* Poluobim */
    float s = (a+b+c)/2;

    return sqrt(s*(s-a)*(s-b)*(s-c));
}

/* Izracunava obim poligona. Argumenti funkcije su niz tacaka
koje predstavljaju temena poligona kao i njihov broj */
float circumference(struct point polygon[], int num)
{
    int i;
    float o = 0.0;

    /* Dodajemo duzine stranica koje spajaju susedna temena */
    for (i = 0; i<num-1; i++)
        o += segment_length(polygon[i], polygon[i+1]);

    /* Dodajemo duzinu stranice koja spaja prvo i poslednje teme */
    o += segment_length(polygon[num-1], polygon[0]);

    return o;
}

/* Izracunava povrsinu konveksnog poligona. Argumenti funkcije su niz tacaka
koje predstavljaju temena poligona kao i njihov broj */
float area(struct point polygon[], int num)
{
    /* Povrsina */
    float a = 0.0;
    int i;
```

```
/* Poligon delimo na trouglove i posebno izracunavamo povrsinu svakoga od njih */
for (i = 1; i < num -1; i++)
    a += Heron(polygon[0], polygon[i], polygon[i+1]);

return a;
}

main()
{
    /* Definisemo dve promenljive tipa tacke */
    struct point a;

    /* Inicijalizujemo tacku b na (1,2) */
    struct point b = {1, 2};

    /* triangle je niz od tri tacke - trougao (0,0), (0,1), (1,0) */
    struct point triangle[3];

    /* square je niz od cetiri tacke - jedinicni kvadrat.
       Obratiti paznju na nacin inicijalizacije niza struktura */
    struct point square[4] = {{0, 0}, {0, 1}, {1, 1}, {1, 0}};

    /* Postavljamo vrednosti koordinata tacke a */
    a.x = 0; a.y = 0;

    /* Gradimo trougao (0,0), (0,1), (1,0) */
    triangle[0].x = 0; triangle[0].y = 0;
    triangle[1].x = 0; triangle[1].y = 1;
    triangle[2].x = 1; triangle[2].y = 0;

    /* Ispisujemo velicinu strukture tacka */
    printf("sizeof(struct point) = %d\n", sizeof(struct point));

    /* Ispisujemo vrednosti koordinata tacaka */
    printf("x koordinata tacke a je %d\n", a.x);
    printf("y koordinata tacke a je %d\n", a.y);
    printf("x koordinata tacke b je %d\n", b.x);
    printf("y koordinata tacke b je %d\n", b.y);

    printf("Obim trougla je %f\n",
           circumference(triangle, 3));
    printf("Obim kvadrata je %f\n",
           circumference(square, 4));
    printf("Povrsina trougla je %f\n",
           Heron(triangle[0], triangle[1], triangle[2]));
    /* Broj tacaka je moguće odrediti i putem sizeof */
    printf("Povrsina kvadrata je %f\n",
           area(square, sizeof(square)/sizeof(struct point)));
}
```

```
}
```

```
Izlaz:
sizeof(struct point) = 8
x koordinata tacke a je 0
y koordinata tacke a je 0
x koordinata tacke b je 1
y koordinata tacke b je 2
Obim trougla je 3.414214
Obim kvadrata je 4.000000
Povrsina trougla je 0.500000
Povrsina kvadrata je 1.000000
```

Primer 5 *Ilustracija korišćenja typedef.*

```
/* Koriscenje typedef radi lakseg rada */
#include <stdio.h>

#include <math.h>
/* Ovim se omogucava da se nadalje u programu umesto int moze
koristiti ceo_broj */
typedef int ceo_broj ;

/* Ovim se omogucuje da se nadalje u programu umesto struct point
moze koristiti POINT */
typedef struct point POINT;

struct point
{
    int x;
    int y;
};

main()
{
    /* Umesto int mozemo koristiti ceo_broj */
    ceo_broj x = 3;

    /* Definisemo promenljivu tipa tacke.
    Umesto struct point mozemo koristiti POINT */
    POINT a;

    printf("x = %d\n", x);

    /* Postavljamo vrednosti koordinata tacke a*/
    a.x = 1; a.y = 2;
    /* Ispisujemo velicinu strukture tacka */
    printf("sizeof(struct point) = %d\n", sizeof(POINT));

    /* Ispisujemo vrednosti koordinata tacaka */
    printf("x koordinata tacke a je %d\n", a.x);
```

```
    printf("y koordinata tacke a je %d\n", a.y);  
}  
  
Izlaz:  
x = 3  
sizeof(struct point) = 8  
x koordinata tacke a je 1  
y koordinata tacke a je 2
```

1.3 Životni vek i oblast važenja promenljivih

Primer 6 *Demonstracija zivotnog veka i oblasti vazenja promenljivih (scope).*

```
#include <stdio.h>  
  
/* Globalna promenjiva */  
int a = 0;  
  
/* Uvecava se globalna promenjiva a */  
void increase()  
{  
    a++;  
    printf("increase::a = %d\n", a);  
}  
  
/* Umanjuje se lokalna promenjiva a. Globalna promenjiva zadrzava svoju vrednost. */  
void decrease()  
{  
    /* Ovo a je nezavisna promenjiva u odnosu na globalno a */  
    int a = 0;  
    a--;  
    printf("decrease::a = %d\n", a);  
}  
  
void nonstatic_var()  
{  
    /* Nestaticke promenjive ne cuvaju vrednosti kroz pozive funkcije */  
    int s=0;  
    s++;  
    printf("nonstatic::s=%d\n",s);  
}  
  
void static_var()  
{  
    /* Staticke promenjive cuvaju vrednosti kroz pozive funkcije.  
    Inicijalizacija se odvija samo u okviru prvog poziva. */  
    static int s=0;
```

```
s++;
printf("static::s=%d\n",s);
}

main()
{
    /* Promenjive lokalne za funkciju main */
    int i;
    int x = 3;

    printf("main::x = %d\n", x);

    for (i = 0; i<3; i++)
    {
        /* Promenjiva u okviru bloka je nezavisna od spoljne promenjive.
           Ovdje se koristi promenjiva x lokalna za blok petlje koja ima
           vrednost 5, dok originalno x i dalje ima vrednost 3*/
        int x = 5;
        printf("for::x = %d\n", x);
    }

    /* U ovom bloku x ima vrednost 3 */
    printf("main::x = %d\n", x);

    increase();
    decrease();

    /* Globalna promenjiva a */
    printf("main::a = %d\n", a);

    /* Demonstracija nestatickih promenljivih */
    for (i = 0; i<3; i++)
        nonstatic_var();

    /* Demonstracija statickih promenljivih */
    for (i = 0; i<3; i++)
        static_var();
}
```

Izlaz iz programa:

```
main::x = 3
for::x = 5
for::x = 5
for::x = 5
main::x = 3
increase::a = 1
decrease::a = -1
main::a = 1
nonstatic::s=1
```

```
nonstatic::s=1  
nonstatic::s=1  
static::s=1  
static::s=2  
static::s=3
```

Primer 7 *Ilustruje vidljivost imena*

```
#include <stdio.h>  
  
int i=10;  
  
void main() {  
    {  
        int i=3;  
        {  
            int i=1;  
            printf("%d\n", i);  
        }  
        printf("%d\n",i);  
    }  
    printf("%d\n",i);  
}
```