

Programiranje 1
Beleške sa vežbi
Školska 2006/2007 godina

Matematički fakultet, Beograd

Jelena Tomašević

December 18, 2006

Sadržaj

1	Programski jezik C	5
1.1	Pokazivači	5
1.2	Prenos parametara po vrednosti i preko pokazivača	7
1.3	Lenjo izračunavanje	8
1.4	Zadaci za vežbu	10

1

Programski jezik C

1

1.1 Pokazivači

Pokazivač je promenljiva koja sadrži adresu promenljive.

```
int x=1, y=1, z[10];
int *ip;    /* ip je pokazivac na int,
             odnosno *ip je tipa int*/

ip = &x;    /* ip sada pokazuje na x */
y=*ip;      /* y je sada 1 */
*ip = 0;     /* x je sada 0 */

*ip+=10;     /* x je sada 10*/
++*ip;       /* x je sada 11*/
(*ip)++;     /* x je sada 12,
             zagrada neophodna zbog prioriteta
             operatora*/

ip = &z[0]; /* ip sada pokazuje na z[0]*/
```

Primer 1 *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);
```

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

```

    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

```

Pored pokazivača na osnovne tipove, postoji i pokazivač na prazan tip (void).

```
void *pp;
```

Njemu može da se dodeli da pokazuje na int, ili na char ili na proizvoljan tip ali je to neophodno eksplicitno naglasiti svaki put kada želimo da koristimo ono na šta on pokazuje.

Primer 2 *Upotreba pokazivača na prazan tip.*

```

#include<stdio.h>

main()
{
    void *pp;
    int x=2;
    char c='a';

    pp = &x;
    *(int *)pp = 17;    /* x postaje 17*/
    printf("\n adresa od x je %p", &x);
    printf("\n%d i %p",*(int*)pp,(int * )pp);

    pp = &c;
    printf("\n adresa od c je %p", &c);
    printf("\n%c i %p",*(char*)pp,(char * )pp);

}

/*
   adresa od x je 0012FF78
   17 i 0012FF78

```

```
    adresa od c je 0012FF74
a i 0012FF74
```

```
*/
```

Posebna konstanta koja se koristi da se označi da pokazivač ne pokazuje na neko mesto u memoriji je NULL.

1.2 Prenos parametara po vrednosti i preko pokazivača

Primer 3 *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
```

C prosleđuje argumente u funkcije pomoću vrednosti. To znači da sledeća funkcija neće uraditi ono što želimo:

```
void swap (int x, int y) /* POGRESNO!!!!!!!!!!*/
{
    int temp;
    temp = x;
    x=y;
    y=temp;
}
```

Zbog prenosa parametara preko vrednosti swap ne može da utiče na argumente a i b u funkciji koja je pozvala swap. Ova swap funkcija samo zamenjuje kopije od a i b.

Da bi se dobio željeni efekat, potrebno je da se proslede pokazivači:

```
/* Zameni *px i *py */
void swap (int *px, int *py)
{
    int temp;
    temp =*px;
    *px = *py;
    *py = temp;
}
```

a poziv funkcije swap izgleda sada ovako

```
swap(&a, &b);
```

Primer 4 *Demonstracija više povratnih vrednosti funkcije koristeći prenos preko pokazivača.*

```
/* Funkcija istovremeno vraća dve vrednosti - količnik i ostatak
dva data broja.
   Ovo se postiže tako što se funkciji predaju vrednosti dva broja (x i y) koji se dele
   i adrese dve promenljive na koje će se smestiti rezultati */
void div_and_mod(int x, int y, int* div, int* mod) {
    printf("Količnik 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 što joj saljemo vrednosti dva broja (5 i 2)
       i adrese promenljivih div i mod na koje će 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 slučaju:
 Adresa promenljive div je 0012FF88
 Adresa promenljive mod je 0012FF84
 Količnik postavljam na adresu : 0012FF88
 Ostatak postavljam na adresu : 0012FF84
 Vrednost promenljive div je 2
 Vrednost promenljive mod je 1

1.3 Lenjo izračunavanje

Primer 5 *Ilustracija lenjog izračunavanja logičkih operatora.*

Prilikom izračunavanja izraza - A && B, ukoliko je A netačno, izraz B se ne izračunava.

Prilikom izračunavanja izraza - A || B, ukoliko je A tačno, izraz B se ne izračunava.

```
#include <stdio.h>

int b = 0;

/* Funkcija ispisuje da je pozvana i uvećava promenljivu b.
   Funkcija uvek vraća vrednost 1 (tačno)
*/
int izracunaj()
{
    printf("Pozvano izracunaj()\n");
    b++;
}
```

```

    return 1;
}

main()
{
    /* Funkcija izracunaj() ce se pozivati samo za parne vrednosti a */
    int a;
    for (a = 0; a < 10; a++)
        if (a%2 == 0 && izracunaj())
            printf("Uslov ispunjen : a = %d, b = %d\n", a, b);
        else
            printf("Uslov nije ispunjen : a = %d, b = %d\n", a, b);

    printf("-----\n");

    /* Funkcija izracunaj() ce se pozivati samo za neparne vrednosti a */
    b = 0;
    for (a = 0; a < 10; a++)
        if (a%2 == 0 || izracunaj())
            printf("Uslov ispunjen : a = %d, b = %d\n", a, b);
        else
            printf("Uslov nije ispunjen : a = %d, b = %d\n", a, b);
}

```

Izlaz:

```

Pozvano izracunaj()
Uslov ispunjen : a = 0, b = 1
Uslov nije ispunjen : a = 1, b = 1
Pozvano izracunaj()
Uslov ispunjen : a = 2, b = 2
Uslov nije ispunjen : a = 3, b = 2
Pozvano izracunaj()
Uslov ispunjen : a = 4, b = 3
Uslov nije ispunjen : a = 5, b = 3
Pozvano izracunaj()
Uslov ispunjen : a = 6, b = 4
Uslov nije ispunjen : a = 7, b = 4
Pozvano izracunaj()
Uslov ispunjen : a = 8, b = 5
Uslov nije ispunjen : a = 9, b = 5
-----
Uslov ispunjen : a = 0, b = 0
Pozvano izracunaj()
Uslov ispunjen : a = 1, b = 1
Uslov ispunjen : a = 2, b = 1
Pozvano izracunaj()
Uslov ispunjen : a = 3, b = 2
Uslov ispunjen : a = 4, b = 2
Pozvano izracunaj()
Uslov ispunjen : a = 5, b = 3
Uslov ispunjen : a = 6, b = 3
Pozvano izracunaj()

```

Uslov ispunjen : a = 7, b = 4
 Uslov ispunjen : a = 8, b = 4
 Pozvano izracunaj()
 Uslov ispunjen : a = 9, b = 5

1.4 Zadaci za vežbu

Zadatak 1 *Sledeći fragment programa obilazi istovremeno nizove sve dok nije $a[i]=b[i]=0$ povećavajući $b[i]$ za 1 svaki put. Da li je program korektan? obrazložiti.*

```
int a[10], b[10];
int i = 0;
...
while(a[i] || b[i]++) i++;
```

Zadatak 2 *Sledeći deo programa obilazi niz t sleva i sdesna istovremeno i zaustavlja se kada su $t[i]$ i $t[j]$ različiti od 0. Da li je program korektan? obrazložiti.*

```
int t[100];
int i=0, j=100;
while(!t[i++] || !t[--j]);
```

Zadatak 3 *Broj je Armstrongov ako je jednak sumi n -tih stepena svojih cifara. Ispitati da li je broj koji se unosi sa standardnog ulaza Armstrongov.*

Zadatak 4 *Za dati broj može se formirati niz tako da je svaki sledeći član niza dobijen kao suma cifara prethodnog člana niza. Broj je srećan ako se dati niz završava sa jedinicom. Napisati program koji za uneti broj određuje da li je srećan.*

Zadatak 5 *Sa ulaza se unosi broj u osnovi deset i osnova ≤ 10 . Odštampati vrednost datog broja u datoj osnovi.*

Zadatak 6 *Sa ulaza se unosi osnova ≤ 10 i broj. Proveriti da li je taj broj ispravan broj za datu osnovu i ako jeste izračunati njegovu vrednost u osnovi 10.*

Zadatak 7 *Broj je Nivenov ako je deljiv sumom svojih cifara.*

1. Napsati funkciju koja računa sumu cifara broja a . Na primer, za broj 121 funkcija treba da vrati 4.
2. Napisati funkciju koja proverava da li je broj Nivenov i vraća 1 ako jeste a 0 ako nije.
3. Napisati program koji za uneto n ispisuje prvih n Nivenovih brojeva.
4. Napisati program koji za uneto n ispisuje sve Nivenove brojeve manje od n .

Zadatak 8 *Napisati program koji izračunava vrednost polinoma u tački x :*

1. Napisati funkciju koja računa k -ti stepen prirodnog broja n .
2. Napisati program koji za uneti niz koeficijenata $a[i]$ i uneti broj x računa vrednost polinoma $a_n * x^n + a_{n-1} * x^{n-1} + \dots + a_1 * x + a_0$