

Osnovi programiranja

Beleške sa vežbi

Smer *Računarstvo i informatika*
Matematički fakultet, Beograd

Jelena Tomašević

December 25, 2005

Sadržaj

1		5
1.1	Funkcije	5
1.2	Lenjo izračunavanje	7

1

1

1.1 Funkcije

Primer 1 *sum* - najjednostavnija funkcija koja sabira dva broja

```
/* Definicija funkcije */
int sum(int a, int b)
{
    return a+b;
}

main()
{
    /* Poziv funkcije */
    printf("%d\n", sum(3,5));
}
```

Primer 2 *Deklaracija funkcije* moze da stoji nezavisno od definicije funkcije. Deklaracija je neophodna u situacijama kada se definicija funkcije navodi nakon upotrebe date funkcije u kodu.

```
int zbir(int, int);

main()
{
    /* Poziv funkcije */
    printf("%d\n", zbir(3,5));
}

/* Definicija funkcije */
int zbir(int a, int b)
{
    return a+b;
}
```

Primer 3 *power* - funkcija koja stepenuje realan broj na celobrojni izlozilac

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

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

```
/* stepenuje x^k tako sto k puta pomnozi x */
int power(float x, int k)
{
    int i;
    float s = 1;
    for (i = 0; i < k; i++)
        s *= x;

    return s;
}

main()
{
    /* Poziv funkcije */
    float s = power(2.0, 8);
    printf("%f\n", s);
}
```

Primer 4 *Verzija koja radi i za negativne izloziocce*

```
int power_n(float x, int k)
{
    int i;
    int negative = k < 0;

    if (negative)
        k = -k;

    float s = 1;
    for (i = 0; i < k; i++)
        s *= x;

    return negative ? 1.0/s : s;
}

main()
{
    /* Poziv funkcije */
    float s = power(2.0, -1);
    printf("%f\n", s);
}
```

Primer 5 *Napisati funkciju koja izračunava zbir n-tih stepena brojeva od 1 do granice i program koji ilustruje rad ove funkcije.*

```
#include <stdio.h>
void Zbir_stepena (int n, int granica);
main()
{
    Zbir_stepena(2, 5);
    Zbir_stepena(3, 5);
}
```

```

    Zbir_stepena(4,10);
    return 0;
}

void Zbir_stepena (int n, int granica)
{
    int i,j;    /*brojaci u for petljama */
    long Zbir=0 , stepenovan ;

    /*spoljasnji for ciklus obavlja sumiranja*/
    for (i=1; i<=granica; Zbir +=stepenovan, ++i)
        /*unutrasnji for ciklus obavlja stepenovanje */
        for( stepenovan=1,j=1; j<=n; stepenovan*= (long) i, ++j) ;
    printf(" Zbir %d. stepena od 1 do %d jeste %ld\n", n,granica,Zbir);
}

```

Izlaz:

```

Zbir 2. stepena od 1 do 5 jeste 55
Zbir 3. stepena od 1 do 5 jeste 225
Zbir 4. stepena od 1 do 10 jeste 25333

```

Primer 6

```

#include <stdio.h>
void Zbir_Kvad(int n);    /*f-ja koja vrši željeno izračunavanje */
main()
{
    Zbir_Kvad( 5);
    Zbir_Kvad( 23);
}

void Zbir_Kvad(int n)
{
    int br; /* lokalna promenljiva funkcije, brojac u ciklusu */
    long Zbir=0; /* lokalna promenljiva funkcije, suma kvadrata brojeva od 1..n */
    for (br=1; br<=n; Zbir+= (long) br*br, ++br) ;
    printf(" Zbir kvadrata brojeva od 1 do %d jese %ld\n", n,Zbir);
}

```

Izlaz:

```

Zbir kvadrata brojeva od 1 do 5 jese 55
Zbir kvadrata brojeva od 1 do 23 jese 4324

```

1.2 Lenjo izračunavanje

Primer 7 *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 uvecava promenjivu b.
   Funkcija uvek vraca vrednost 1 (tacno)
*/
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

```

Primer 8 *Napisati program u C-u koji prikazuje sve proste brojeve u datom intervalu kojima je zbir cifara složen broj. Interval se zadaje učitavanjem gornje i donje granice (dva prirodna broja). Brojeve prikazati u opadajućem poretaku.*

```

#include <stdio.h>
#include <stdlib.h>

int prost (int n); /*testira da li je broj n prost broj */
/*Prirodni brojevi (sem 1)imaju najmanje dva delioca:jedinicu i samog sebe.
Brojevi koji nemaju drugih delioca,sem ova dva, nazivaju se prostim */

int zbirCifara (int n); /*vraca zbir cifara broja n */
main()
{
    int donja,gornja; /*granice intervala */
    int i; /*brojac u petlji */
    int pom; /*posrednik u eventualnoj zameni */

    /*ucitavanja granice intervala */
    scanf("%d%d", &donja, &gornja);
    if (donja > gornja) /*obezbedjivanje relacije: donja <=gornja */
    {
        pom=donja;
        donja=gornja;
        gornja=pom;
    }
    for(i=gornja;i>=donja; i--)
        if (prost (i) && !prost(zbirCifara(i) ) ) printf("%d\n",i);
}

int prost(int n)
/*Ispituje se da li je broj n prost tako to se proverava da li ima delioce
medju brojevima od 2 do n/2. Pri implementaciji se koristi tvrdjenje da je
broj prost ako je jednak 2, ili ako je neparan i ako nema delitelja medju
neparnim brojevima od 3 do n/2 */
{
    int prost; /*indikator slozenosti broja n */
    int i; /*potencijalni delitelj broja n */
    if (n==1) return 0;

```

```

    /*parni brojevi razliciti od  od dva nisu prosti brojevi */
    prost= (n%2!=0) || (n==2);

    /*najmanji potencijalni kandidat za delitelje medju
    neparnim brojevima razlicitim od jedan */
    i=3;
    while ( (prost) && (i<=n/2) )
    {
        prost=n%i != 0;
        i=i+2; /*proveravamo kandidate za delitelje samo medju neparnim brojevma */
    }
    return prost;
}
int zbirCifara (int n)
{ int Suma=0;
  while (n>0)
  {
      Suma+= n%10; /*dodavanje cifre tekuceg razreda,pocev od razreda jedinica ,
                  a iduci ka visim razredima cifara */
      n=n/10;      /*prelaz ka visem razredu */
  }
  return Suma;
}

```

Ulaz:

1 20

Izlaz:

19

17

13

Zadaci za vežbu:

Zadatak 1 Napisati f-ju koja za uneti broj n izračunava zbir recipročnih vrednosti prvih n brojeva.

Zadatak 2 Ilustracija korišćenja funkcije za izračunavanje faktoriijela celog broja.

(a) Napisati program koji izračunava faktoriijel unetog broja.

(b) Napisati funkciju koja izračunava faktoriijel celog broja.

(c) Napisati program koji izračunava faktoriijel unetog broja koristeći prethodno definisanu funkciju.

Zadatak 3 Ilustracija korišćenja funkcije za proveru da li je broj prost.

(a) Napisati program koji za uneti broj proverava da li je prost.

(b) Napisati funkciju koja za ceo broj proverava da li je prost.

(c) Napisati program koji štampa prvih 100 prostih brojeva.