

Programiranje 1  
*Beleške sa vežbi*  
*Školska 2006/2007 godina*

Matematički fakultet, Beograd

Jelena Tomašević

December 12, 2006



# Sadržaj

|          |                                    |          |
|----------|------------------------------------|----------|
| <b>1</b> | <b>Programski jezik C</b>          | <b>5</b> |
| 1.1      | Nizovi — osnovni pojmovi . . . . . | 5        |
| 1.2      | Funkcije . . . . .                 | 7        |



# 1

## Programski jezik C

1

### 1.1 Nizovi — osnovni pojmovi

Deklaracija niza:

```
int niz[5]; /* niz od 5 elemenata tipa int*/
```

Pristupanje elementima niza:

```
niz[0] = 4;
niz[1] = 2 * niz[0];      /*niz[1] = 8*/
niz[2] = niz[0] * niz[1]; /*niz[2] = 32*/
niz[3] = 5;
niz[4] = 7;
```

Unos vrednosti elemenata niza sa tastature:

```
for(i=0; i<5; i++)
    scanf("%d ", &a[i]);
```

Stampanje elemenata niza

```
for(i=0; i<5; i++)
    printf("%d ", a[i]);
```

Brojanje elemenata niza je od nule!

Indeks niza može da bude proizvoljan izraz celobrojne vrednosti: `niz[i*2]=5`.

**Primer 1** Program ilustruje korišćenje statičkih nizova. Ispisuje 10 unetih brojeva unazad.

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

```
main()
{
    int a[10];
    int i;
    for (i = 0; i<10; i++)
```

---

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

```
{    printf("a[%d]=",i);
    scanf("%d",&a[i]);
}

printf("Unazad : \n");

for (i = 9; i>=0; i--)
    printf("a[%d]=%d\n",i,a[i]);
}
```

**Primer 2** *Brojanje pojavljivanja svake od cifara. Koriscenje niza brojača.*

```
#include <stdio.h>
#include <ctype.h>
main()
{
    /* Niz brojaca za svaku od cifara */
    int br_cifara[10];
    int i, c;

    /* Resetovanje brojaca */
    for (i = 0; i < 10; i++)
        br_cifara[i] = 0;

    /* Citamo sa ulaza i povecavamo odgovarajuce brojace */
    while ((c = getchar()) != EOF)
        if (isdigit(c))
            br_cifara[c-'0']++;

    /* Ispis rezultata */
    for (i = 0; i < 10; i++)
        printf("Cifra %d se pojavila %d put%s\n",
            i, br_cifara[i], br_cifara[i]==1?"":"a");
}
```

**Primer 3** *Program ilustruje inicijalizaciju nizova.*

```
#include <stdio.h>

main()
{
    /* Niz inicijalizujemo tako sto mu navodimo vrednosti
       u viticasnim zagradama. Dimenzija niza se odredjuje
       na osnovu broja inicijalizatora */
    int a[] = {1, 2, 3, 4, 5, 6};

    /* Isto vazi i za niske karaktera */
    char s[] = {'a', 'b', 'c'};

    /* Ekvivalentno prethodnom bi bilo
    char s[] = {97, 98, 99};
```

```
*/

/* Broj elemenata niza */
int a_br_elem = sizeof(a)/sizeof(int);
int s_br_elem = sizeof(s)/sizeof(char);

/* Ispisujemo nizove */

int i;
for (i = 0; i < a_br_elem; i++)
    printf("a[%d]=%d\n", i, a[i]);

for (i = 0; i < s_br_elem; i++)
    printf("s[%d]=%c\n", i, s[i]);

}
```

## 1.2 Funkcije

**Primer 4** *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 5** *Deklaracija funkcije može 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 6** *power* - funkcija koja stepenuje realan broj na celobrojni izlozilac

```
#include <stdio.h>

/* 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 7** *Verzija koja radi i za negativne izlozioc*e

```
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 8** *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 sumiranje*/
    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 9** *Napisati funkciju koja izračunava zbir kvadrata brojeva od 1 do date granice kao i program koji ilustruje korišćenje date funkcije.*

```

#include <stdio.h>
void Zbir_Kvad(int n);    /*f-ja koja vrši željeno izracunavanje */
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

```

**Primer 10** *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 poretku.*

```

#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 2 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 brojevima */
    }
    return prost;
}

int zbirCifara (int n)
{ int Suma=0;
  while (n>0)
  {
    Suma+= n%10; /*dodavanje cifre tekuceg razreda,pocев 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** Sa tastature učitati elemente niza koji su celi brojevi i za koje se pretpostavlja da ih nema više od 100. Pronaći maksimalan elemenat niza i ispisati ga na izlaz.

**Zadatak 2** Sa tastature se unosi 15 karaktera u niz. Ispitati da li uneti niz predstavlja palindrom (primer palindroma je "anavolimilovana").

**Zadatak 3** Ispisati prvih 15 članova Fibonačijevog niza.

**Zadatak 4** Napisati program koji ispituje da li dva niza imaju barem jedan zajednički element.

**Zadatak 5** Napisati f-ju koja za uneti broj n izračunava zbir recipročnih vrednosti prvih n brojeva.

**Zadatak 6** 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 7** 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.