

PROGRAMIRANJE 1

PROGRAMSKI JEZIK

C

Staša Vujičić
Čas 9

FUNKCIJE

- Veliki računski zadaci mogu se razbiti u manje delove i time se omogućava ljudima da iskoriste ono što su neki drugi već uradili, umesto da počinju sve od početka. Odgovarajuće funkcije skrivaju detalje postupka od delova programa i time čine ceo program jasnijim i jednostavnijim za menjanje.
- Prilikom deklaracije funkcije navodi se
 - tip povratne vrednosti funkcije (ako se ne navede podrazumeva se int)
 - ime funkcije
 - lista argumenata
 - telo funkcije.
- Funkcija se poziva navođenjem imena funkcije i liste stvarnih argumenata.

FUNKCIJE

- ◉ *Napisati funkciju koja vrši sabiranje dva cela broja i program koji testira rad ove funkcije.*

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

```
/* Definicija funkcije */
```

```
int zbir(int a, int b)
```

```
{
```

```
    return a+b;
```

```
}
```

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

- ◉ 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.

```
#include <stdio.h>
/* Deklaracija funkcije zbir() */
    int zbir(int, int); /* int zbir(int a, int b); */

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

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

FUNKCIJE

- Nalaženje maksimuma dva broja

Program pronalazi maksimum dva broja koji se učitavaju sa ulaza. Maksimum se izračunava u posebnoj funkciji. Program demonstrira upotrebu najjednostavnijih funkcija.

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

```
/* Funkcija max() prihvata dva broja  
i vraca njihov maksimum */
```

```
int max(int a, int b)
```

```
{
```

```
    return a > b ? a : b;
```

```
}
```

```
/* Funkcija main */  
int main()  
{  
    int a, b, m;  
  
    /* Ucitavamo dva cela broja */  
    printf("Uneti dva broja: ");  
    scanf("%d%d", &a, &b);
```



```
/* Pozivamo funkciju max(), predajemo joj  
   vrednosti promenljivih a i b, i njenu  
   povratnu vrednost smestamo u m */
```

```
m = max(a,b);
```

```
/* Prikazujemo m */
```

```
printf("Maksimum je: %d\n", m);
```

```
}
```

FUNKCIJE

- *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 vrsi zeljeno  
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+=br*br, ++br) ;
```

```
printf(" Zbir kvadrata brojeva od 1 do %d je  
       %ld\n", n,Zbir);
```

```
}
```

FUNKCIJE

○ Ojlerova funkcija

Program računa vrednost Ojlerove funkcije pozitivnog celog broja unetog sa ulaza.

Pod Ojlerovom funkcijom $\varphi(n)$ podrazumevamo broj brojeva m , takvih da je $1 \leq m < n$ i da su m i n uzajamno prosti.

Ojlerova funkcija, kao i NZD(m, n) se računaju u posebnim funkcijama. Program demonstrira upotrebu funkcija.

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

```
/* Deklaracije funkcija (prototipovi) */
```

```
int nzd(int n, int m);
```

```
int euler(int n);
```

```
/* main */
```

```
int main()
```

```
{
```

```
    int n;
```

```
/* Unosimo ceo broj, a zatim izracunavamo  
njegovu Ojlerovu funkciju, i ispisujemo  
vrednost na izlaz */
```

```
printf("Uneti pozitivan ceo broj: ");
```

```
scanf("%d", &n);
```

```
printf("euler(%d) = %d\n", n, euler(n));
```

```
return 0;
```

```
}
```

```
/* Ojlerova funkcija broja n */
int euler(int n)
{
    int br = 0, m;

    /* Prolazimo kroz sve prirodne brojeve m koji su
    manji od n, i za svaki koji je uzajamno prost sa n
    uvecavamo brojac */
    for(m = 1; m < n; m++)
        if(nzd(n,m) == 1)
            br++;

    /* Vracamo broj uzajamno prostih brojeva
    sa brojem n */
    return br;
}
```

```
/* Funkcija racuna nzd dva pozitivna broja primenom  
euklidovog algoritma. */
```

```
int nzd(int n, int m)
```

```
{
```

```
    int r;
```

```
    /* Dokle god je  $m \neq 0$ , svodimo  $nzd(n,m)$  na  
     $nzd(m,r)$ , gde je  $r = n \% m$ . */
```

```
    while(m)
```

```
    {
```

```
        r = n % m;
```

```
        n = m;
```

```
        m = r;
```

```
    }
```

```
    /* Nakon sto je m postalo 0, jasno je da je  $nzd(n,0) =$   
     $n$  */
```

```
    return n;
```

```
}
```


FUNKCIJE

- ◉ Stepen broja

Program računa n -ti stepen unetog realnog broja. Računanje ovog stepena se realizuje u posebnoj funkciji. Učitavanje i ispis se takođe izdvajaju u posebne funkcije.

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

```
/* Funkcija nema argumente, i ne vraca  
vrednost. Jedina njena uloga je da ispise  
pozdravnu poruku */
```

```
void ispisi_poruku()
```

```
{  
    printf("Program racuna n-ti stepen realnog  
    broja!\n");  
}
```

```
/* Funkcija ucitava sa ulaza realan broj,  
   i vraca ga kao povratnu vrednost.  
   Funkcija nema argumente */
```

```
double ucitaj_realan_broj()
```

```
{
```

```
    double a;
```

```
    printf("Uneti realan broj: ");
```

```
    scanf("%f", &a);
```

```
    return a;
```

```
}
```

```
/* Funkcija ucitava sa ulaza ceo broj,  
   i vraca ga kao povratnu vrednost.  
   Funkcija nema argumente */
```

```
int ucitaj_ceo_broj()  
{  
    int a;  
    printf("Uneti ceo broj: ");  
    scanf("%d", &a);  
    return a;  
}
```

```
/* Funkcija vraca n-ti stepen broja a */  
double stepen(double a, int n)  
{  
    double r;  
    for(r = 1; n ; n--, r *= a);  
  
    return r;  
}
```

/* Funkcija prihvata realan broj, i prikazuje ga na izlazu, sa tri decimale. Funkcija ne vraca vrednost */

```
void prikazi_broj(double a)
{
    printf("%5.3f\n", a);
}
```

```
int main()
{
    int n;
    double a;

    ispisi_poruku(); /* Prikaz pozdravne poruke */
    a = učitaj_realan_broj(); /* Učitava realan broj */
    n = učitaj_ceo_broj(); /* Učitava ceo broj */
    prikazi_broj(stepen(a, n)); /* Racuna stepen i
    ispisuje rezultat */

    return 0;
}
```

FUNKCIJE

- Stepen broja

Program računa n -ti stepen unetog realnog broja, gde broj n može biti negativan. Računanje ovog stepena se realizuje u posebnoj funkciji.


```
double power_k(double x, int k)
{
    int i;
    int negative = k<0;

    if (negative)
        k = -k;
    double s = 1;
    for (i = 0; i<k; i++)
        s*=x;
    return negative ? 1.0/s : s;
}
```

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

FUNKCIJE - VEŽBA

- ◉ (a) *Napisati funkciju koja izračunava faktorijel celog broja.*
- ◉ (b) *Napisati program koji izračunava faktorijel unetog broja koristeći prethodno definisanu funkciju.*