

Programiranje i programski jezici



Funkcije

- Razbijaju velike računarske zadatke u manje delove
- Skrivaju detalje postupka od delova programa koji ne moraju da znaju o njima
- Čine program jasnijim i jednostavnijim za menjanje

Struktura programa

- Programi se najčešće sastoje od više manjih funkcija
- Program se može nalaziti u jednoj ili više izvornih datoteka
- Izvorne datoteke se mogu kompajlirati odvojeno i od njih se može formirati izvršni program

Definicija funkcije

- Svaka definicija funkcije ima oblik

```
tip_rezultata naziv_funkcije( deklaracije argumenata )  
{  
    deklaracije i naredbe  
}
```

- Prazna funkcija

```
prazna() {}
```

- Ako tip rezultata nije naveden podrazumeva se **int**
- Kod funkcija koje ne vraćaju vrednost kao tip rezultata se navodi **void**
- Naredba **return** se koristi za vraćanje vrednosti pozivaocu. Vrednost može biti bilo koji izraz. Tip izraza se pretvara u tip rezultata ako je neophodno.

```
return izraz
```

- Pozivalac može ignorisati vraćenu vrednost

Eksterne promenljive

- Interne promenljive se definišu unutar funkcija
- Eksterne promenljive se definišu van funkcija i samim tim su dostupne svim funkcijama
- Funkcije su uvek eksterne zato što C ne dozvoljava definisanje funkcije unutar druge funkcije
- Eksterne promenljive su globalno dostupne i mogu se koristiti umesto argumenata ili vrednosti koja se vraća
- Ukoliko funkcije razmenjuju veliki broj parametara, nekada je pogodnije da to čine preko eksternih promenljivih. Ipak, to nije preporučljivo jer narušava strukturu programa i može dovesti do neželjenih efekata
- Za razliku od automatskih promenljivih koje se stvaraju prilikom ulaska u funkciju i uništavaju prilikom izlaska iz nje, eksterne promenljive su trajne

Zadatak 1

Napisi program koji će za uneti ceo broj izračunati njegov faktorijel koristeći funkciju za računanje faktorijela.

```
#include <stdio.h>
long fakt(int x);
main()
{
    int x;
    printf("Unesi broj"); scanf("%d",&x);
    printf("Njegov faktorijel je %ld\n",fakt(x));
}
long fakt(int x)
{
    long i,res=1;
    for (i=1;i<x; )res=res*(++i);
    return(res);
}
```

Zadatak 2

Napisi program koji će izračunati sume $\sum_{i=1}^5 i^2$ i $\sum_{i=1}^{23} i^2$ i obe sume ispisati u zasebnim linijama.

```
#include <stdio.h>
void Zbir_Kvad(int n);
main()
{
    Zbir_Kvad(5);
    Zbir_Kvad(23);
}
void Zbir_Kvad(int n) {
    int br;
    long Zbir=0;
    for(br=1; br<=n; Zbir+= (long)br*br,++br);
    printf(" Zbir kvadrata brojeva od 1 do %d je %ld\n", n,Zbir);
}
```

Zadatak 3

Napisati program koji će izračunati i ispisati na standardni izlaz

sume: $\sum_{i=1}^5 i^2$, $\sum_{i=1}^5 i^3$, $\sum_{i=1}^{10} i^4$

Zadatak 3

```
#include <stdio.h>
void Zbir_stepena (int n, int k); /*izracunava sumu k-tih stepena
za brojeve od 1 do n */
main()
{
    Zbir_stepena(5,2);
    Zbir_stepena(5,3);
    Zbir_stepena(10,4);
}
void Zbir_stepena (int n, int k)
{
    int i,j;
    long Zbir=0, stepenovan ;
    for (i=1; i<=n; Zbir +=stepenovan, ++i)
    for( stepenovan=1,j=1; j<=k; stepenovan*= (long) i, ++j) ;
    printf(" Zbir %d. stepena od 1 do %d jeste %ld\n", k,n,Zbir);
}
```

Zadatak 4

Napisati program koji ispisuje prvih n prostih brojeva.

Zadatak 4

```
#include<stdio.h>
int prost(int);
main()
{
    int n,i,br;
    printf("Unesite koliko prostih brojeva zelite da dobijete: \n");
    scanf("%d", &n);
    i = 0;br = 2;
    while(i < n) {
        if (prost(br)) {
            printf("Broj %d je prost.\n", br);
            i++;
        }
        br++;
    }
}
```

Zadatak 4

```
int prost(int n)
{
    int prost,i;
    if (n==1) return 0;
    prost= (n%2!=0) || (n==2);
    i=3;
    while ((prost) && (i*i<=n)){
        prost=n%i != 0;
        i=i+2;
    }
    return prost;
}
```

Zadatak 5

Napisati program koji sadrži f-ju g koja za argument **zn** tipa karakter vraća ceo broj po sledećem pravilu:

$$g(zn) = \begin{cases} 1, & zn \text{ je malo slovo} \\ -1, & zn \text{ je veliko slovo} \\ 0, & \text{inace} \end{cases}$$

U glavnom programu se unose karakteri dok se ne unese znak za novi red, korišćenjem f-je **g** se izračunava i na kraju ispisuje koliko je od unetih karaktera bilo malih, a koliko velikih slova.

Zadatak 5

```
#include <stdio.h>
int g(char);
main(){
    int n,i,m=0,v=0;
    char c;
    while ((c=getchar())!='\n'){
        if (g(c)==1) m++;
        else if (g(c)==-1) v++;
    }
    printf("Malih slova ima %d, a velikih %d",m,v);
}
int g(char zn){
    if ((zn>='a') && (zn<='z')) return 1;
    else if ((zn>='A') && (zn<='Z')) return -1;
    else return 0;
}
```