

Laboratorijska vježba 5

1. Napisati program koji za dva unijeta broja na izlazu ispisuje njihov zbir. Program demonstrira rad sa funkcijama.

/* Pisanje funkcije kojoj se predaju parametri po vrijednosti i koja preko svog imena vraća vrijednost */

```
#include <stdio.h>
float suma(float a,float b){
float z;
z=a+b;
return z; /* funkcija preko imena vraca "float" vrijednost */
}

main(){
float a,b,c;
a=10.25; //uzimamo proizvoljno dva broja
b=20.50;
printf("\n Definisani su brojevi a=%f i b=%f ",a,b);
c=suma(a,b);
printf("\n\n Suma brojeva a i b je: %f",c);
getch();
}
```

Ili:

```
#include <stdio.h>
float suma(float a,float b){
return a+b;
}

main(){
float a,b;
a=10.25; //uzimamo proizvoljno dva broja
b=20.50;
printf("\n Definisani su brojevi a=%f i b=%f ",a,b);
printf("\n\n Suma brojeva a i b je: %f", suma(a,b)); /* (ugnjezdavanje) */
getch();
}
```

2. Napisati program koji sadrži funkciju koja vraća vrijednost većeg od dva broja.

```
#include <stdio.h>
int max(int x, int y);

main( ){
int a, b, c;
printf("\nZadati dva cela broja: "); scanf("%d%d", &a, &b);
c = max(a, b);
if(c == 0) printf("\nZadati brojevi su isti!!!\n");
else printf("\nVeci od dva zadata broja je broj %d\n\n", c);
}
```

```

getch();
}

int max(int x, int y){
if(x == y) return 0;
else return ((x>y)?x:y);
}

```

3. Napisati program koji sadrži funkciju koja vraća vrijednost najvećeg od četiri prirodna broja.

```

int max(int x, int y){
return ((x>=y)?x:y);
}

int max4 (int x, int y, int z, int w){
return (max(max(x,y), max(z,w)));
}

main( ){
int a, b, c, d;
printf("\nZadati cetiri cela broja: ");
scanf("%d%d%d%d", &a, &b, &c, &d);
printf("\nNajveci od cetiri zadata broja je %d\n\n", max4(a,b,c,d));
getch();
}

```

4. Napisati program koji sadrži funkciju za izračunavanje faktoriijela unijetog broja. Definisati prototip funkcije. Program prikazuje pisanje funkcije koja uzima argument i vraća vrijednost preko svog imena.

```

#include<stdio.h>
long faktorijel(short); /* prototip funkcije koja vraća long */

main(){
short n;
printf("Unesite mali cijeli broj cijij faktorijel zelite da izracunate.");
printf("\n\n n= ");
scanf("%d",&n);
printf("\n\nFaktorijel broja n=%d je %lu",n,faktorijel(n));
getch();
}

long faktorijel(short n){
long rezultat=1;
int i;
for(i=1 ;i<=n;++i)
rezultat*=i;
return rezultat;
}

```

5. Uraditi prethodni zadatak korišćenjem rekurzije.

Rekurzivna funkcija je svaka funkcija koja poziva samu sebe. Ukoliko u nekom problemu otkrijemo rekurzivnost, to znači da se naš posao može razbiti u korake na taj način, da svaki korak bude sastavljen od dva dijela: (1) jednog nezavisnog i (2) jednog koji izgleda baš kao i početni posao. Tada se nezavisni dio izračuna, a za ostatak posla pozove se ponovo sama funkcija. Npr. :

$$n! = n * (n-1) * (n-2) * \dots * 3 * 2 * 1$$

Ako grupišemo članove na ovaj način:

$$n! = n * [(n-1) * (n-2) * \dots * 3 * 2 * 1]$$

Tada je:

$$n! = n * (n-1)!$$

I to je baš forma koja nam je trebala! Za $n!$ treba izračunati $(n-1)!$, što je posao (2), te to pomnožiti sa n što je posao (1). Kada se u ovom slučaju prestaje sa rekurzivnim pozivima?

Npr.

$$3! = 3 * 2! \dots \text{imamo rekurzivni poziv}$$

$$2! = 2 * 1! \dots \text{imamo rekurzivni poziv}$$

$$1! = 1 \text{ nema rekurzije}$$

Iz priloženog primjera vidimo da rekurzivni pozivi prestaju onog trena kada argument funkcije postane jednak 1, tj. matematika nam govori i da je $0!$ isto tako jednako 1, pa možemo reći da rekurzivni pozivi prestaju za 1 ili za 0.

```
#include<stdio.h>
long faktorijel(short); /* prototip funkcije koja vraća long */

main(){
short n;
printf("Unesite mali cijeli broj čiji faktorijel želite da izračunate.");
printf("\n\n n= ");
scanf("%d",&n);
printf("\n\nFaktorijel broja n=%d je %lu",n,faktorijel(n));
getch();
}

long faktorijel(short n){
if(n>1) return n*faktorijel(n-1);
else return 1;
}
```

6. Napisati program koji sadrži funkciju koja izračunava najveći zajednički djelilac dva prirodna broja (NZD) koristeći Euklidov algoritam.

Napomena:

U matematici, **najveći zajednički djelilac** (NZD) dva cijela broja je najveći pozitivan cijeli broj koji dijeli oba broja bez ostatka.

Zadatak rješavamo pomoću Euklidovog algoritma:

- Ako je $a=b$, tada je $NZD=a$ i to je kraj algoritma.
- Od većeg broja oduzmemo manji i vraćamo se na prvi korak.

```
#include<stdio.h>
int NZD(int a, int b)
{
while(a!=b) {
if (a>b)
a-=b;
else
b-=a;
}
return a;
}

main(){
int a, b,c;
printf("Unesite dva prirodna broja\n");
scanf("%d %d",&a,&b);
printf("NZD je: %d\n",NZD(a,b));
getch();
}
```