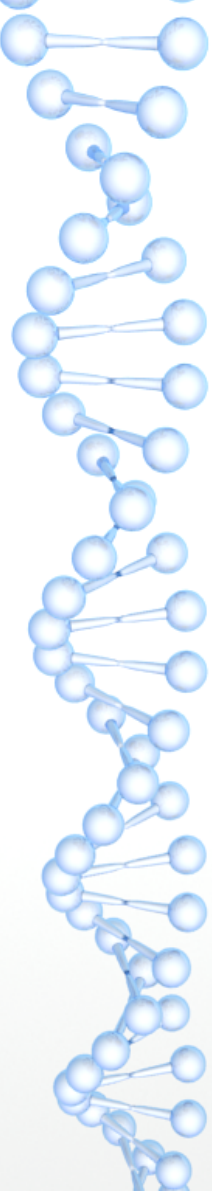


Funkcije

- **Funkcija** je skup ili grupa iskaza koji zajedno kao jedna celina obavljaju neki zadatak
- **Definicija** funkcije se sastoji iz zaglavlja (header) i tela (body) funkcije, pri čemu se zaglavlje funkcije sastoji iz povratne vrednosti, naziva funkcije i liste parametara.
- **Deklaracija** funkcije ukazuje kompajleru dve bitne informacije o funkciji a to su ime i način na koji se funkcija poziva
- **Pozivanje** funkcije u C-u se ostvaruje navođenjem naziva funkcije iza koga sledi lista stvarnih parametara



Načini pozivanja funkcija

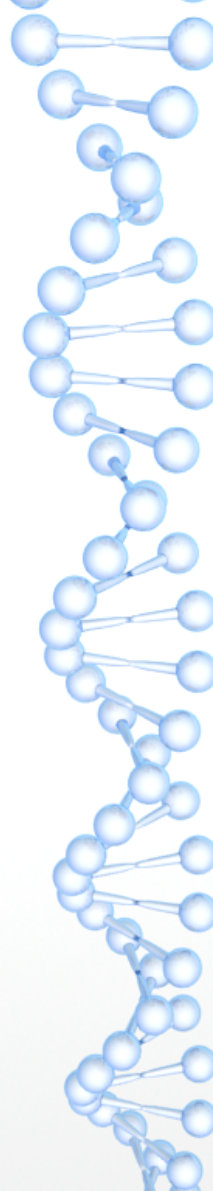
- U C-u poziv funkcije može biti po vrednosti i adresi
- **Formalni parametri funkcije** se tretiraju kao lokalne promenljive koje žive sve dok postoji i funkcija koja ih koristi, i imaju prednost u odnosu na globalne promenljive sa istim nazivom
- **Poziv funkcije po vrednosti** – izvorni kod unutar funkcije radi lokalno i ne može da izmeni vrednost argumenata koji se prosleđuju funkciji
- **Poziv funkcije po adresi** – unutar funkcije se koristi adresa da bi se pristupilo stvarnom parametru, što znači da se izmene vrše nad stvarnim argumentom
-

6.5 Nizovi kao argumenti funkcije

- Predaja argumenata
 - Niz se predaje kao argument funkcije tako što se preda ime niza bez “uglastih” zagrada

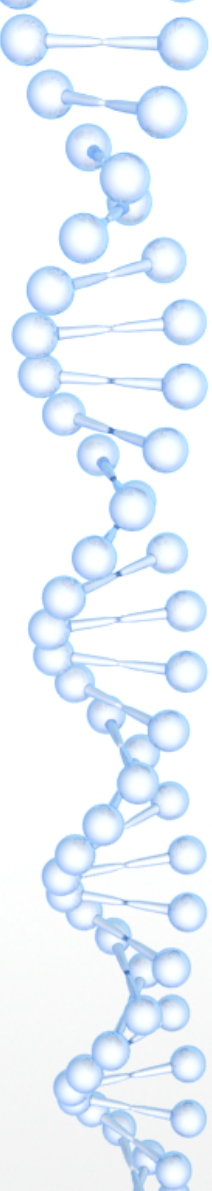
```
int myArray[ 24 ];  
myFunction( myArray, 24 );
```

 - Veličina niza se obično predaje kao drugi argument
 - Nizovi se predaju po referenci
 - Ime niza je adresa prvog elementa
 - Funkcija zna gdje je niz smješten u memoriji
 - Mijenja se originalna memorijska lokacija
- Elementi niza kao argumenti
 - Predaju se po vrijednosti
 - Predaje se kao obična promjenljiva (na primjer, `myArray[ 3 ]`)



Povratne vrednosti funkcija

- **Vrednost:** Kada se rezultat funkcija **vraća po vrednosti**, kopija rezultata se vraća pozivaocu, a ovaj metod funkcionise na isti način kao poziv po vrednosti
- **Adresa (pokazivač)** – Povratak po adresi može vratiti samo adresu promenljive, i ništa drugo. Najčešće se koristi da se vrati novo-zauzeta memorijska adresa.



Oblast definisanosti promenljivih

- Oblast definisanosti je region programa u kome je definisana promenljiva i u kome živi, i van koga ne može biti vidljiva niti dostupna.
- **Lokalne promenljive** su one promenljive koje su definisane u okviru bloka izraza i prestaju da postoje kada prestaje blok u kome su definisane
- **Globalne promenljive** žive tokom celog životnog veka programa i može im se pristupiti iz bilo koje funkcije definisane u okviru programa

ARGUMENT FUNKCIJE JE POKAZIVAČ

Napisati funkciju koja istovremeno vraća dve vrednosti - količnik i ostatak dva data broja. Kao argumente funkcije koristiti pokazivače umesto običnih vrednosti

```
#include <stdio.h>

void Kolicnik_I_Ostatak(int x, int y, int* kolicnik, int* ostatak)
{
    printf("Kolicnik se postavlja na adresu : %p\n", kolicnik);
    printf("Ostatak se postavlja na adresu : %p\n", ostatak);
    *kolicnik = x / y;
    *ostatak = x % y;
}

void main()
{
    int kolicnik, ostatak;

    printf("Adresa promenljive kolicnik je: %p\n", &kolicnik);
    printf("Adresa promenljive ostatak je: %p\n", &ostatak);

    Kolicnik_I_Ostatak(5, 2, &kolicnik, &ostatak);

    printf("Vrednost kolicnika je: %d\n", kolicnik);
    printf("Vrednost ostatka je: %d\n", ostatak);
}
```