

## 29. Radna nedelja (06. 04. 2020. - 10.04. 2020.)

Predmet : Računari i programiranje

Odeljenje: III2

Nastavna jedinica : Funkcije, Prenos parametara po vrednosti

Nastavnik : Ivan Mladenović

**Odgovore na pitanja i rešenja zadatka poslati do 12.04.2020 god. na e-mail**

[ivan.mladenovic9901@gmail.com](mailto:ivan.mladenovic9901@gmail.com)

### ARGUMENTI FUNKCIJE - PRENOS PARAMETARA PO VREDNOSTI

Parametri se funkciji prenose po vrednosti, što znači da funkcija zapravo radi sa njihovim lokalnim kopijama. Sve promene koje funkcija načini nad parametrima, zapravo se odnose na lokalne kopije parametara, a ne na same parametre.

Zaključak je da se preneti parametri NE MOGU promeniti u telu funkcije.

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

```
void f (int x)
```

```
{
```

```
x *= 2;
```

```
x++;
```

```
}
```

```
int main()
```

```
{
```

```
int x = 3;
```

```
f(x);
```

```
printf("%d\n", x);
```

```
return 0;
```

```
}
```

#### ***Analiza prethodnog primera***

U prethodnom primeru, biće ispisano 3, jer se promene na x-u, vrše nad njegovom lokalnom kopijom u funkciji f, a ne baš nad x-om.

•Ako hoćemo da u funkciji promenimo x, jedan od načina je da iz funkcije vratimo to promenjeno x (na kraju funkcije da stavimo return x;), dok u main funkciji da kažemo x = f(x);

#### ***Analiza prenosa parametara u primeru sa funkcijom zbir***

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

```
* Definicija funkcije */
```

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

```
int main() {
```

```
/* Poziv funkcije */
```

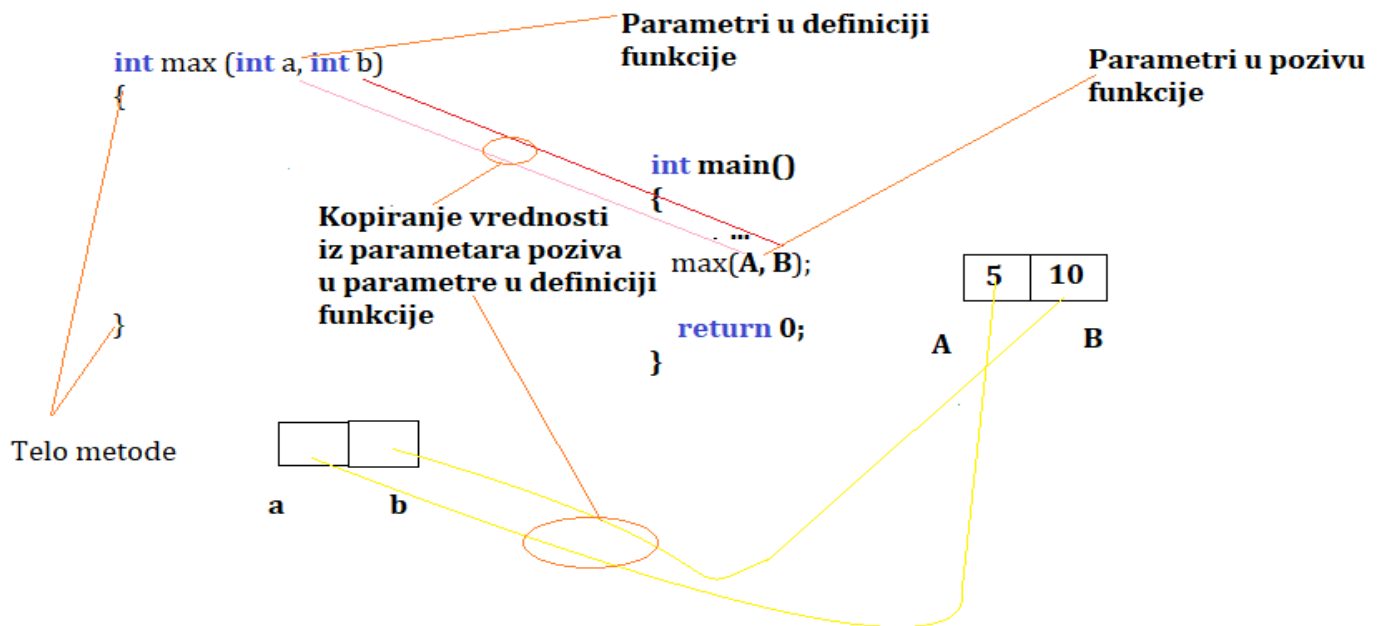
```
cout<< zbir(3,5);
```

```
}
```

*Kada se u main funkciji pojavi poziv funkcije rezultat = zbir (5,3) realizuju se sledeće akcije:*

- Rezerviše se memorijski prostor za promenljive deklarisanе u funkciji zbir();
- Formalnim parametrima se dodeljuju vrednosti stvarnih parametara: a=3; b=5;
- Izvršava se telo funkcije;
- Rezultat izračunavanja u funkciji se postavlja na mesto obraćanja toj funkciji, odnosno dodeljuje se promenljivoj rezultat, i prelazi se na izvršenje sledeće naredbe u funkciji main().

Na slici je predstavljeno šta se događa kada pozovemo funkciju ( kada joj prosledimo argumente na slici „parametri u pozivu“ ). Ovo je primer prosleđivanja po vrednosti.



Znači prilikom poziva se parametri prosleđuju funkciji ( kopiraju se na neku lokaciju i na toj adresi se izvršavaju naredbe funkcije nad podacima ( kopiranim podacima ) stvarni podaci koje smo poslali iz main a ostaju nepromenjeni na svojoj lokaciji.

Koja je mana ovog načina pozivanja (prosleđivanja parametara)

Kad bi parametri koje prosleđujemo nekoj funkciji bili veličine nekoliko stotina MB pa i GB a ne kao u primeru brojevi, prilikom pozivanja tj kopiranja parametara došlo bi do značajnog zauzeća memorije( parametere koje šaljemo kopiramo na drugu lokaciju tj dupliramo).

## Zadaci :

1. Upisati broj i ispitati je li prost. ( koristiti funkciju koja ispituje da li je broj prost).
2. Upisati 10 brojeva. Ispisati koliko je upisano parnih brojeva , a koliko neparnih ?

( Napraviti funkciju koja broji koliko ima parnih elemenata u nizu I funkciju koja broji neparne elemente  
Niz kao parameter I argument pr: ( funkcija sabirak ima parameter niz s     **int sabirak( int s[10])**

Kako pozvati ovu funkciju ( treba joj proslediti niz )

```
int main()
{
    int niz[10];
    for (int i = 0; i < 10; i++)
    {
        cin >> niz[i];
    }
    cout << "broj parnih je " << sabirak(niz);
}
```

3. Upisati dva broja i ispisati veći koristeći pokazivače. ( zadatak uraditi bez pravljenja funkcije)

## Pitanja:

1. Šta su parametri a šta argumenti funkcije?
2. Objasniti postupak predavanja parametra funkciji ( prilikom poziva) ,šta se dešava sa podatkom koji prosleđujemo funkciji I koja vrednost se vraća I gde ( ovo za slučaj da prosleđujemo parameter po vrednosti)?