



OSNOVE PROGRAMIRANJA U C-U

Privatni časovi Novi Sad

Skripta za polaznike kursa

Sadržaj

Glava 1. Uvod u materijal	2
Glava 2. Uvod u C	3
Osnovni pojmovi	3
O programskom jeziku C	3
Prvi program u C-u	4
Rezervisane reči	6
Osnovne aritmetičke operacije	7
Zadaci za samostalan rad	8

Glava 1. Uvod u materijal

Kome je skripta namenjen?

Skripta iz osnova programiranja namenjen je svima koji nemaju nikakvo predznanje iz programiranja, ili imaju vrlo skromnu osnovu. U njoj su objedinjeni materijali sa vežbi i predavanja na različitim smerovima FTN-a i PMF-a, kao i sa raznih sajtova i kurseva programiranja. Zadaci su objašnjeni na vrlo jednostavan način, tako da neko ko nema iskustva s programiranjem može da isprati o čemu se radi. Skripta se koristi kao dodatni materijal za samostalan rad studenata kod kuće i kao pomoćni materijal za sve koji žele da nauče osnove programiranja kod nas. Za zadatke iz prvih nekoliko glava korišćen je Code::Blocks. Zadaci iz naprednih lekcija rade se na virtuelnoj mašini, u Linux-u.

Šta ćemo naučiti?

Obrađuje se sledeći spisak lekcija:

- Prvi program u C-u. Tipovi podataka, osnovne računske operacije, čitanje i ispis podataka.
- If kontrolna struktura. Switch-case naredbe. Složene operacije i izrazi.
- Rad sa petljama. For, while, do-while petlja.
- Nizovi i stringovi (opciono: višedimenzioni nizovi).
- Rad sa funkcijama. Organizacija koda, izdvajanje koda u funkcije, stilizovanje koda.
- Pokazivači.
- Rad sa fajlovima. Čitanje i ispis podataka. Korišćenje githuba.
- Liste. Smeštanje podataka iz fajlova u liste.
- Rekurzija.
- Stabla –binarna stabla. Dinamička alokacija memorije.
- Osnovni algoritmi – sortiranje nizova i listi.

*Moguće su manje izmene lekcija od strane predavača, kao i obrada dodatnih tema ukoliko ostane vremena na časovima.

Glava 2. Uvod u C

Osnovni pojmovi

Algoritam = detaljan opis potrebnih koraka, operacija i njihovog redosleda izvršavanja, potrebnih da bi se obradili podaci

Programski jezik = formalni jezik na kom računaru objašnjavamo šta treba da radi

O programskom jeziku C

C je proceduralni programski jezik opšte namene. Razvio ga je Denis Riči 70tih godina prošlog veka u Belovim laboratorijama da bi redizajnirao operativni sistem Unix, odnosno da bi omogućio njegovo korišćenje na različitim računarima. Pre pojave C-a korišćen je assembler za potrebe izrade operativnih sistema. Za C kažemo da je programski jezik nižeg nivoa, što znači da radi s istim objektima kao i sama mašina (računar). To su brojevi, znakovi (npr. +, - ...) i adrese. C pravi razliku između velikih i malih slova, što znaci da za njega „prviSabirak“ i „prvisabirak“ nisu iste promenljive. Pogodan je za razvoj operativnih sistema i kompajlera.

Korisne osobine:

- Prenosivost izvornog koda –kodovi nisu zavisni od arhitekture mašine na kojoj ih pišemo
- Efikasnost
- C je jednostavan jezik za učenje
- Može da se koristi i za programiranje na niskom nivou (pisanje skripti za drajvere i kernele) i za programiranje na visokom nivou (softverske aplikacije)

I mnoge druge...

Prvi program u C-u

Započecemo sa analizom jednostavnog primera .

Primer 1. Napisati program koji učitava 3 broja, a zatim računa njihovu aritmetičku sredinu. Algoritam po kome se izvršava ovaj program u uprošćenom obliku izgleda ovako:

1. Učitaj od korisnika 3 broja
2. Izračunaj zbir ova tri broja
3. Podeli sa 3 sumu da bi dobio aritmetičku sredinu
4. Ispiši rezultat

Naš kod izgleda kao na slici ispod.

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

int main()
{
    int prviSabirak;
    int drugiSabirak;
    int treciSabirak;
    int suma;
    float aritmetickaSredina;
    printf("Unesite prvi sabirak \n");
    scanf("%d",&prviSabirak);
    printf("Unesite drugi sabirak \n");
    scanf("%d",&drugiSabirak);
    printf("Unesite treći sabirak \n");
    scanf("%d",&treciSabirak);
    suma = prviSabirak+drugiSabirak+treciSabirak;
    aritmetickaSredina = suma /3;
    printf("aritmeticka sredina je %f",aritmetickaSredina);

    return 0;
}
```

Nakon pokretanja koda, rezultat koji dobijamo je:

```
C:\Users\Marina\Desktop\systemClitratura\Program\bin\Debug\Program.exe
Unesite prvi sabirak
3
Unesite drugi sabirak
4
Unesite treci sabirak
5
aritmeticka sredina je 4.000000
Process returned 0 (0x0)   execution time : 19.066 s
Press any key to continue.
```

Mogli smo da napišemo naš kod i na sledeći način:

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

int main()
{
    int prviSabirak, drugiSabirak, treciSabirak, suma;
    float aritmetickaSredina;
    printf("Unesite sabirke \n");
    scanf("%d %d %d", &prviSabirak, &drugiSabirak, &treciSabirak);
    suma = prviSabirak+drugiSabirak+treciSabirak;
    aritmetickaSredina = suma /3;
    printf("aritmeticka sredina je %f", aritmetickaSredina);

    return 0;
}
```

Sada ćemo videti detaljno šta radi jedan po jedan red u našem programu.

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

int main()
{
    int prviSabirak, drugiSabirak, treciSabirak, suma;
    float aritmetickaSredina;
    printf("Unesite sabirke \n");
    scanf("%d %d %d", &prviSabirak, &drugiSabirak, &treciSabirak);
    suma = prviSabirak+drugiSabirak+treciSabirak;
    aritmetickaSredina = suma /3;
    printf("aritmeticka sredina je %f",aritmetickaSredina);

    return 0;
}
```

Prva dva reda u našem programu predstavljaju „uvoz biblioteka“. Pretprocesorska direktiva *#include* služi da u naš kod uključi datoteke koje je neko drugi pisao i omogući nam korišćenje tuđeg koda. Funkcije *printf()* i *scanf()* koje koristimo za učitavanje i ispisivanje podataka nalaze se u biblioteci *<stdio.h>*.

<stdlib.h> nam za sada nije neophodan, ali ćemo je uključivati u svaki naš program.

Na sledećim linkovima možete videti šta se sve nalazi u ovim bibliotekama.

https://www.tutorialspoint.com/c_standard_library/stdio_h.htm

https://www.tutorialspoint.com/c_standard_library/stdlib_h.htm

Pored ove dve biblioteke, u našim zadacima ćemo često koristiti i *<math.h>*.

https://www.tutorialspoint.com/c_standard_library/math_h.htm

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

int main()
{
    int prviSabirak, drugiSabirak, treciSabirak, suma;
    float aritmetickaSredina;
    printf("Unesite sabirke \n");
    scanf("%d %d %d", &prviSabirak, &drugiSabirak, &treciSabirak);
    suma = prviSabirak+drugiSabirak+treciSabirak;
    aritmetickaSredina = suma /3;
    printf("aritmeticka sredina je %f",aritmetickaSredina);

    return 0;
}
```

Program u jeziku C se sastoji od jedne ili više funkcija. Funkcije su međusobno ravnopravne i mogu pozivati jedna drugu. Izvršavanje C programa počinje od funkcije *main()* . Iz nje se mogu pozivati sve ostale funkcije. Svaki program mora imati jednu *main()* funkciju. Najčešće ima povratnu vrednost tipa int. Sam pojam funkcije će kasnije biti detaljnije objašnjen, za sada treba znati da funkcija ima svoje ime, može imati jednu povratnu vrednost i može imati nula ili više ulaznih argumenata (parametara).

Napomena: pošto je *main()* funkcija (ne naredba), neophodne su zagrade iza njenog naziva!

Funkcija *main()* obično sadrži blok naredbi koji počinje posle otvorene vitičaste zagrade { i nastavlja se do zatvorene vitičaste zagrade }. Većina naredbi završava se sa ; .


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

int main()
{
    int prviSabirak, drugiSabirak, treciSabirak, suma;
    float aritmetickaSredina;

    printf("Unesite sabirke \n");
    scanf("%d %d %d", &prviSabirak, &drugiSabirak, &treciSabirak);
    suma = prviSabirak+drugiSabirak+treciSabirak;
    aritmetickaSredina = suma /3;
    printf("aritmeticka sredina je %f",aritmetickaSredina);

    return 0;
}
```

Na početku bloka naredbi deklariramo varijable(promenljive) koje ćemo koristiti u zadatku. Kada deklariramo promenljive, u memoriji ćemo zauzeti mesto na koje će se smestiti učitani podatak. Svaka promenljiva ima svoj tip i ime. Sintaksa je

<tip promenljive > <ime promenljive>;

Na osnovu tipa promenljive računar zna koliko memorije treba da zauzme, i koje operacije će moći sa njom da izvodi.

Naša memorija:

prviSabirak:	<input type="text"/>
drugiSabirak:	<input type="text"/>
treciSabirak:	<input type="text"/>
suma:	<input type="text"/>
aritmetickaSredina:	<input type="text"/>

Za ovaj primer potrebne su nam 4 celobrojne promenljive koje predstavljaju sabirke i sumu, i jedna realna promenljiva koja predstavlja njihovu aritmetičku sredinu.

Celobrojni tip označava se sa *int*, dok se realan tip podataka označava sa *float*. U

narednoj tabeli dat je pregled tipova podataka koji postoje i koliko memorije svaki od njih zauzima.

DATA TYPE	MEMORY (BYTES)	RANGE	FORMAT SPECIFIER
short int	2	-32,768 to 32,767	%hd
unsigned short int	2	0 to 65,535	%hu
unsigned int	4	0 to 4,294,967,295	%u
int	4	-2,147,483,648 to 2,147,483,647	%d
long int	4	-2,147,483,648 to 2,147,483,647	%ld
unsigned long int	4	0 to 4,294,967,295	%lu
long long int	8	$-(2^{63})$ to $(2^{63})-1$	%lld
unsigned long long int	8	0 to 18,446,744,073,709,551,615	%llu
signed char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
float	4		%f
double	8		%lf
long double	12		%Lf

Nakon što smo deklarirali promenljive , sledeći korak je da korisniku ispišemo poruku šta treba da uradi.

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

int main()
{
    int prviSabirak, drugiSabirak, treciSabirak, suma;
    float aritmetickaSredina;
    printf("Unesite sabirke \n");
    scanf("%d %d %d", &prviSabirak, &drugiSabirak, &treciSabirak);
    suma = prviSabirak+drugiSabirak+treciSabirak;
    aritmetickaSredina = suma /3;
    printf("aritmeticka sredina je %f", aritmetickaSredina);

    return 0;
}
```

Printf() funkcija služi za ispisivanje teksta na ekran. Ono što navedemo između navodnika biće ispisano korisniku. Kao što možete primetiti u primeru, rečenica se završava sa „\n“, ali taj simbol nakon pokretanja programa neće biti ispisano. On govori vašem programu da nakon što ispiše „Unesite sabirke“, treba da pređe u novi red.

Sintaksa za *printf()* naredbu ima opšti oblik :

printf ("Ovo je tekst koji hoćemo da ispišemo \n ");

Napomena: Simbol „\n“ nije obavezan deo, ali je preporučljivo da ga koristimo zbog preglednosti ispisa.

Naredna naredba je poziv *scanf()* funkcije, koja nam služi da učitamo vrednosti sa tastature.

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

int main()
{
    int prviSabirak, drugiSabirak, treciSabirak, suma;
    float aritmetickaSredina;
    printf("Unesite sabirke \n");
    scanf("%d %d %d", &prviSabirak, &drugiSabirak, &treciSabirak);
    suma = prviSabirak+drugiSabirak+treciSabirak;
    aritmetickaSredina = suma /3;
    printf("aritmeticka sredina je %f",aritmetickaSredina);

    return 0;
}
```

Sintaksa za `scanf()` naredbu u opštem obliku izgleda na sledeći način:

```
scanf(„%skraćena_oznaka_tipa_podatka",&ime_promenljive_koju_učitavamo);
```

Kada naiđe na ovu funkciju računar prekine sa radom i sačeka da korisnik unese neku vrednost i pritisne *enter*. Simbol & je neophodan pre naziva promenljive, jer nam on daje adresu u memoriji na kojoj je promenljiva smeštena (više reči o ovome biće u poglavlju sa pokazivačima). Nakon što korisnik pritisne *enter*, računar unosi unetu vrednost u promenljivu koju smo joj namenili. U našem primeru uneli smo 3, 4 i 5. Obratite pažnju da u navedenom primeru istovremeno učitavamo vrednost sva tri sabirka. U tom slučaju opšti oblik sintakse izgleda kao:

```
scanf(„%skraćena_oznaka_1tipa %skracena_oznaka_2tipa ...",&ime1_promenljive,
&ime2_promenljive,...);
```

Naša memorija sada ovako izgleda:

prviSabirak:	<input type="text" value="3"/>
drugiSabirak:	<input type="text" value="4"/>
treciSabirak:	<input type="text" value="5"/>
suma:	<input type="text"/>
aritmetickaSredina:	<input type="text"/>

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

int main()
{
    int prviSabirak, drugiSabirak, treciSabirak, suma;
    float aritmetickaSredina;
    printf("Unesite sabirke \n");
    scanf("%d %d %d", &prviSabirak, &drugiSabirak, &treciSabirak);

    suma = prviSabirak+drugiSabirak+treciSabirak;
    aritmetickaSredina = suma /3;
    printf("aritmeticka sredina je %f",aritmetickaSredina);

    return 0;
}
```

Sledeća naredba je naredba dodele vrednosti. Računar sabira elemente sa desne strane jednakosti, i upisuje dobijeni rezultat u promenljivu na levoj strani jednakosti. Jednostavno rečeno: u promenljivu *suma* računar upisuje zbir unetih sabiraka. Nakon toga, promenljivoj *aritmetickaSredina*, dodeljuje vrednost sume podeljene sa tri.

Stanje u memoriji nakon računanja:

prviSabirak:	<input type="text" value="3"/>
drugiSabirak:	<input type="text" value="4"/>
treciSabirak:	<input type="text" value="5"/>
suma:	<input type="text" value="12"/>
aritmetickaSredina:	<input type="text" value="4"/>

Paznja: Simbol „/“ služi za celobrojno deljenje i zbog toga ćemo kao rezultat za unos 1, 2 ,2 dobiti 1 . O ovome će biti više reči kasnije!

```
#include <stdio.h>

#include <stdlib.h>

int main()
{
    int prviSabirak, drugiSabirak, treciSabirak, suma;
    float aritmetickaSredina;
    printf("Unesite sabirke \n");
    scanf("%d %d %d", &prviSabirak, &drugiSabirak, &treciSabirak);
    suma = prviSabirak+drugiSabirak+treciSabirak;
    aritmetickaSredina = suma /3;
    printf("aritmeticka sredina je %f", aritmetickaSredina);

    return 0;
}
```

Na kraju, ponovo pozivamo *printf()* naredbu, koja sada ima nešto drugačiji oblik. Ukoliko želimo da ispišemo vrednost promenljive na ekran, potrebno je da pod navodnicima umetnemo **%oznaka_tipa** na mesto gde želimo da vrednost bude ispisana. Primer:

```
Broj_godina=10;
```

```
Adresa = 5;
```

```
printf („Živim %d godina u Novom Sadu, na adresi Kraljevića Marka %d“, broj_godina,  
adresa);
```

ispisaće:

Živim **10** godina u Novom Sadu, na adresi Kraljevića Marka **5**.

Kao što vidimo, na mesto gde se nalazi prvi %d program će ispisati 10, a na mesto drugog %d ispisaće 5.

Poslednja naredba u našoj main() funkciji je poziv *return 0*, koja završava naš program i govori nam da se on uspešno izvršio.

Ako, ipak, postoji greška koji je prevodilac(kompajler) detektovao, onda će se ispisati poruka o grešci uz referencu na linije u kojima u izvornoj datoteci postoji greška.

Rezervisane reči

Rezervisane reči opisuju neke specijalne konstrukcije /naredbe programa. Kada pišemo kod ne smemo našim promenljivim i funkcijama davati imena koja predstavljaju rezervisane reči.

Spisak rezervisanih reči:

auto	double	int	struct	break	else	long	switch	case
enum	register	typedef	char	extern	return	union	const	
float	short	unsigned	continue	for	signed	void	default	
goto	sizeof	volatile	do	if	static	while		

Osnovne aritmetičke operacije

Sada ćemo kroz jedan primer videti osnovne operacije u C-u.

Primer 2. Napisati program koji učitava dva broja, a zatim računa njihov zbir, razliku, proizvod, celobrojni količnik, realan količnik i ostatak pri deljenju.

Rešenje:

```
#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()
{
    int a = 3;
    int b = 5;

    printf("Zbir a+b je : %d\n",a+b);
    printf("Razlika a-b je : %d\n",a-b);
    printf("Proizvod a*b je : %d\n",a*b);
    printf("Celobrojni kolicnik a/b je : %d\n", a/b);
    printf("Pogresan pokusaj racunanja realnog kolicnika a/b je : %f\n", a/b);
    printf("Realni kolicnik a/b je : %f\n", (float)a/(float)b);
    printf("Ostatak pri deljenju a/b je : %d\n", a%b);
    printf("Koren broja a je : %f\n", sqrt(a));
    printf("Koren broja a ispisan na 2 decimale je : %.2f\n", sqrt(a));

    return 0;
}
```

Od osnovnih aritmetičkih operacija imamo **+**, **-**, *****, **/** koje nam daje rezultat celobrojnog deljenja, osim ukoliko mu ne naglasimo da želimo realan količnik. Dakle, ako računamo $3 \setminus 5$ rezultat celobrojnog deljenja je 0, a ostatak je 3. Da bismo kao rezultat dobili 0.6 što jeste prava vrednost nakon deljenja, nije dovoljno samo da stavimo da nam je izlazni rezultat `%f`, kao što je u redu:

`printf("Pogresan pokusaj racunanja realnog kolicnika a/b je : %f\n", a/b);`

Potrebno je da „peške“ konvertujemo *a* i *b* u realne brojeve, što postizemo dodavanjem (**float**) ispred samih promenljivih u izrazu.

Još jedna novina u ovom odeljku je izraz **%f** – on govori vašem programu da realan rezultat treba da zaokruži na dve decimale.

Pitanja:

Šta će ispisivati program ako za tip promenljivih *a* i *b* postavimo *float* umesto *int*?

Šta će se desiti ako kod računanja realnog količnika izbacimo (*float*) ispred *a* ili ispred *b*?

Zadaci za samostalan rad

1. Učitati sa standardnog ulaza dužine stranica pravougaonika. Zatim izracunati obim i površinu.
2. Učitati sa standardnog ulaza temperaturu u Kelvinima i Konvertovati u Celzijuse. Veza između ove dve temperature data je formulom:
$$T_k = T_c + 273.15$$
3. Učitati sa standardnog ulaza dužine stranica trougla, zatim izračunati obim i površinu koristeći Heronov obrazac.

$$\text{Heronov obrazac : } s = \frac{a+b+c}{2}$$
$$P = \sqrt{s * (s - a) * (s - b) * (s - c)}$$

- 4*. Napisati program koji računa rastojanje između tačaka na osnovu njihovih koordinata.

$$\text{Rastojanje između 2 tačke: } d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- 5*. Napisati program koji od korisnika zahteva unos tri broja, a zatim računa njihovu harmonijsku, geometrijsku i kvadratnu sredinu.

$$\text{Harmonijska sredina : } \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

$$\text{Geometrijska sredina: } \sqrt[n]{x_1 * x_2 * x_3 \dots * x_n}$$

$$\text{Kvadratna sredina: } \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}}$$

6* .Date su koordinate dva naspramna temena pravougaonika čije su stranice paralelne koordinatnim osama. Napisati program kojim se prikazuju dužina dijagonale, obim i površina tog pravougaonika.