

Osnovi programiranja



2024/25



Ulazno-izlazne naredbe

- Razmenu podataka između programa i standardnog ulaza (tastatura, fajl na disku, ...) i izlaza (ekran, štampač, disk, ...) možemo izvršiti korišćenjem ulazno-izlaznih naredbi.
- Ulazna naredba programskog jezika C za učitavanje sa tastature je **scanf**, a izlazna naredba za ispisivanje na ekran je **printf**.
- U programskom jeziku C postoje i druge varijante ovih naredbi, namenjene za učitavanje iz datoteka i stringova, kao i za štampanje u datoteke i stringove.

Ulazne naredbe

- Kada u programu deklarišemo određene promenljive, time smo samo obezbedili memorijsku lokaciju za smeštanje njihovih vrednosti.
- Naredba za učitavanje vrednosti sa standardnog ulaza u programskom jeziku C je **scanf**.
- Ova naredbe preuzima vrednosti sa standardnog ulaza i dodeljuju ih odgovarajućim promenljivama.

```
scanf("%format_unosa", &naziv_promenljive);
```

Format unosa	Tip promenljive
%d	int
%u, %o, %x	unsigned int
%f	float
%lf	double
%c	char
%s	niz karaktera (String)

Ulazne naredbe

- Ukoliko promenljivama i i j gde je i ceo broj, a j realan broj, treba dodeliti vrednosti 41 i 19.6 sa tastature, onda bi naredba izgledala ovako:

```
scanf(“%d%f“, &i, &j);
```

- Za razdvajanje brojeva prilikom unosa mogu se koristiti jedan ili više praznih mesta (taster SPACE), znak za tabulator (taster TAB) ili znak za kraj reda (taster ENTER).
- Ukoliko se vrši učitavanje podataka tipa **char**, naredba **scanf** prazna mesta ne tretira kao separatore, već kao ravnopravne znakove.

```
scanf(“%f%c%c“, &a, &c1, &c2);    2.33E+2 Ana
```

```
a=233.0
```

```
c1= ' '
```

```
c2= 'A'
```

Ulazne naredbe

- Promenljive tipa string se učitavaju tako što naredba **scanf** očitava sve karaktere do razmaka.

```
char rec1[10], rec2[5];  
scanf("%s", rec1);  
scanf("%s", rec2);
```

- Podaci logičkog tipa se mogu učitati kao podaci tipa **int**, ali njihova vrednost ne može biti manja od nule.

Izlazne naredbe

- Nakon obrade podataka u programu, potrebno je rezultate prikazati korisniku ili ih zabeležiti na nekom medijumu.
- Da bi se podaci ispisali na ekranu koristi se naredba programskog jezika C printf, koja vrši ispisivanje navedenih podataka na standardnom izlazu.
- Pored ispisa navedenih podataka, korišćenjem ove naredbe moguće je definisati i format u kome se ti podaci ispisuju, kako bi prikaz bio čitljiviji za korisnika.

```
printf(“%d %d“, 1, 2);  
printf(“%d“, 1);  
printf(“\n“);  
printf(“+ %d\n = 3“, 2);
```

Izlazne naredbe

- Nakon obrade podataka u programu, potrebno je rezultate prikazati korisniku ili ih zabeležiti na nekom medijumu.
- Da bi se podaci ispisali na ekranu koristi se naredba programskog jezika C printf, koja vrši ispisivanje navedenih podataka na standardnom izlazu.
- Pored ispisa navedenih podataka, korišćenjem ove naredbe moguće je definisati i format u kome se ti podaci ispisuju, kako bi prikaz bio čitljiviji za korisnika.

```
printf(“%d %d“, 1, 2);  
printf(“%d“, 1);  
printf(“\n“);  
printf(“+ %d\n = 3“, 2);
```

Specijalne sekvence

- `\n` novi red
- `\t` tabulator
- `\b` povratnik (backspace)
- `\"` navodnik
- `\\` za obrnutu crtu (backslash)

Argumenti funkcije printf -opisi formata konverzije (iza znaka %)

Pravila:

- %c je za štampanje karaktera
- %3c je za štampanje karaktera u polju širine tri pozicije
- %d štampaj kao označen ceo dekadni broj
- %6d štampaj kao ceo broj u polju širine 6 pozicija
- %f štampaj kao realan broj
- %6f štampaj kao realan broj u polju širine 6
- %o štampaj kao oktalni broj
- %x štampaj kao heksadekadni broj
- %.2f štampaj kao realan broj sa dve decimale
- %6.2f štampaj kao realan broj u polju širine 6 ynakova, a od toga 2 iza decimalne tačke
- %s štampaj string
- %% za štampanje procenta

Primer 1

- Šta je rezultat datog koda?

```
#include<stdio.h>
main()
{
    printf("Prvi primer "
           "je ovo\n");
}
```

Primer 2

- Šta je rezultat datog koda?

```
#include <stdio.h>
main()
{
    int a=21,d=5;
    float c=8.953, b=6.0;
    printf("d=%d c=%f\n",d,c);
    printf(" |%5f |%5.3f| %5.0f|%5.1f\n",c,c,c,c);
}
```

Primer 3

- Šta je rezultat datog koda?

```
#include <stdio.h>
main()
{
    int x = 428;
    double y = -8475.54;
    double z = 5.645e+03;
    printf("%8d \n", x);
    printf("%10.2lf \n", y);
    printf("%5d%5d \n", x, x);
}
```

```
      428
-8475.54
 428  428
```

Primer 4

- Šta je rezultat datog koda?

```
#include<stdio.h>
#define PI 3.141592
main()
{
    float obim;
    float r=2.5;
    obim=2*r*PI;
    printf("Obim je: %f\n",obim);
}
```

Primer 5

- Šta je rezultat datog koda?

```
#include<stdio.h>
#define RED “prvi\ndrugi\ntreci”
main()
{
    printf(“%s\n”,RED);
}
```

Primer 6

- Šta je rezultat datog koda?

```
#include <stdio.h>
main()
{
    printf("Sabiranje celih brojeva:%d\n", 4+10);
    printf("Oduzimanje celih brojeva:%d\n", 4-10);
    printf("Deljenje celih brojeva:%d\n", 15/2);
    printf("Deljenje celih brojeva:%d\n", 2/4);
    printf("Deljenje celih brojeva:%d\n", -7/2);
    printf("Ostatak pri deljenju:%d\n", 15%2);
}
```

Primer 7

- Šta je rezultat datog koda?

```
#include <stdio.h>
main()
{
    int x=506, y=3, z=21, t=2;
    printf("Kolicnik z / t =%d\n",z / t);
    printf("Kolicnik z / t =%f\n",z / t);
    printf("Kolicnik z / t =%f\n",(float)z / t);
    printf("-x=%d\n",-x);
    printf("Ostatak pri deljenju x %% y=%d\n", x%y);
}
```


Primer 8

- Napisati program kojim računar simulira bacanje kockice, tj. generiše slučajan ceo broj na intervalu od 1 do 6.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
main()
{
    srand(time(0));
    printf("Broj : %d" ,rand()%6+1);
}
```

srand se zove samo jedanput kao generator da ne bi svaki poziv rand funkcije generisao istu sekvencu

rand daje broj iz intervala [0, RAND_MAX]. RAND_MAX se razlikuje od računara do računara ali nije manji od 32767.

Primer 9

- Napisati program koji sa standardnog ulaza učitava realan broj i ispisuje na standardni izlaz prvu decimalu (prvu cifru iza decimalne tacke).