

## Redoslijed obavljanja aritmetičkih operacija i konverzija tipova podataka

Kolika je vrijednost varijabli i, j i x, i u kojoj se domeni obavlja računanje izraza za varijable i, j i x?

```
int main() {
    int i, j;
    float x;
    i = (int) (2.5 * 6) % 3 + 2.6;
    j = i * 3 / 4;
    x = (float) (j * 5 / 2);
    printf ("%d %d %f", i, j, x);
    return 0;
}
```

Moguća pitanja: zašto treba eksplicitni cast `(int) (2.5 * 6)`. Odgovor jer operator `%` radi samo nad cjelobrojnim operandima a `2.5 * 6` se računa u double domeni. Zašto `x` nije 2.5 a napred je float cast `(float) (j * 5 / 2)`; Odgovor: zato što se izračunavanje već obavilo u int domeni jer su svi operandi cjelobrojni.

Rješenje je: 2 double, 1 integer, 2.0 float

## Konstantni niz znakova

Koliko okteta (bajtova) se koristi za pohranu sljedećih konstanti napisanih u programskom jeziku C:

```
1
"Tekst\n//"
'1'
1.
"1.33"
'\x55'
"\\//"
"0"
```

Rješenje:

4 + 9 + 1 + 8 + 5 + 1 + 4 + 2

## Logički tip podataka, relacijski operatori, logički izrazi i naredbe

Koje će vrijednosti imati varijable x, y i z nakon izvođenja sljedećeg odsječka?

```
int a, b, c, d, x, y, z;
a = 10; b = 20; c = 30; d = 40;
x = a < b || c < b && 0;
y = a < b && c > c;
z = ! a == b;
```

Rješenje: x=1 y=0 z=0

Napomenuti da paze na prednost && pred || (kod varijable x). Pokazati da bi dobili drugačije rješenje drugim redoslijedom. Paziti na prednost negacije (kod varijable z). Na primjerima pokazati što znači !a (!5, !0, !(-1)).

## Logički izrazi s logičkim operatorima

Riješite se operatora negacije koristeći de Morganova pravila:

`! (x > 20 && x < 30 || x > 100)`

Rješenje:      `= !(x > 20 && x < 30) && !(x > 100)`  
                 `= (! (x > 20) || ! (x < 30)) && !(x > 100)`  
                 `= ((x <= 20) || (x >= 30)) && (x <= 100)`

## Jednostrana selekcija

Napisati program koji učitava tri cijela broja i ispituje jesu li prvi i drugi broj djeljivi s trećim bez ostatka, te ispisuje odgovarajuću poruku. Treći broj ne smije biti 0.

```
#include <stdio.h>
int main() {
    int a, b, c;

    printf("Unesite a, b i c:");
    scanf("%d %d %d", &a, &b, &c);
    if (c != 0) {
        if (a % c == 0) {
            printf("%d jest djeljiv s %d\n", a, c);
        }
        if (b % c == 0) {
            printf("%d jest djeljiv s %d\n", b, c);
        }
    }
    return 0;
}
```

## Jednostrana selekcija – složeniji uvjet

Učitati tri cijela broja. Ispisati odgovarajuću poruku, ako je prvi učitani broj ujedno i najveći. Ako je prvi broj najveći, ispitati je li jednak ili umnošku ili zbroju druga dva broja, pa ako jest, ispisati i umnožak i zbroj druga dva broja.

```
int main () {
    int n, a, b;
    scanf ("%d", &n);
    scanf ("%d", &a);
    scanf ("%d", &b);
    if (n > a && n > b) {
```

```

    printf("Prvi upisani broj je najveći!\n");
    if (n == a * b || n == a + b)
        printf ("%d %d", a * b, a + b);
}
return 0;
}

```

## Raspon realnih brojeva

Za pohranu realnih brojeva koristi se registar u kojem mantisa **zajedno sa skrivenim bitom** ima 51 bit, karakteristika ima 12 bita, te se jedan bit koristi za predznak.

Koliko približno iznosi najveći realni broj kojeg je moguće prikazati u registru?

*Napomena: svi posebni slučajevi se prikazuju analogno standardu IEEE 754.*

Rješenje:

Karakteristika može prikazati  $2^{12} = 4096$  različitih vrijednosti:

- vrijednost 0 je rezervirana za prikaz broja 0
- vrijednost 4095 je rezervirana za prikaz  $\infty$
- ostaje 4094 različitih vrijednosti što čini raspon binarnih eksponenata: [-2046, 2047]

Dakle, najveći broj je:

$$1.11111111 \dots * 2^{2047} \approx 2 * 2^{2047} \approx 2^{2048}$$

## ASCII tablica i operacije sa znakovnim tipom podataka

Učitati znak. Ukoliko se radi o slovu, ispisati redni broj tog slova u abecedi.

```

#include <stdio.h>
int main () {
    char a, b;
    scanf ("%c", &a);

    if (a >= 65 && a <= 90)
        printf ("Slovo %c - redni broj: %d", a, a-64);
    else if (a >= 97 && a <= 122)
        printf ("Slovo %c - redni broj: %d", a, a-96);
    return 0;
}

```

## Apsolutne, relativne i numeričke pogreške

1. Što treba učiniti kako bi se u prikazu broja prema IEEE 754 formatu jednostruke preciznosti, najveća moguća **relativna pogreška** prepolovila (tj. smanjila 2 puta)?

Rješenje: povećati mantisu za jedan bit.

2. Najveća moguća **apsolutna** pogreška koja se može očekivati pri prikazu realnog broja 0.001 u IEEE 754 formatu **jednostruke** preciznosti jest približno:

Rješenje:  $6 \cdot 10^{-11}$

### Tema: Cjelobrojne i realne konstante i varijable u programskom jeziku C

Navedite koje od sljedećih definicija varijabli nisu ispravne. Kao točan odgovor priznavat će se samo onaj u kojem su nabrojane SVE neispravne definicije.

- 1) float tall = 9.7e11F;
- 2) double int x = 91278127;
- 3) long high = 03L;
- 4) long int 7\_a = 0x7AU;
- 5) double char y = 15;

Rješenje: 2, 4, 5

Što će se na zaslon ispisati sljedećim programom?

```
#include <stdio.h>
int main () {
    float x;
    unsigned char c;
    int i;
    x = 0x30f;
    c = 127U + 2;
    i = 012;
    printf("%f %d %d", x, c, i);
    return 0;
}
```

Rješenje: 48.000000 129 10

## Dvostrana selekcija

Napisati program koji će za učitana tri cijela broja odrediti koji je najveći.

```
#include <stdio.h>
int main()
{
    int  a, b, c;
    int  max;

    scanf("%d %d %d", &a, &b, &c);
    if( a > b ) {
        if( a > c )
            max = a;
        else
            max = c;
    }
    else {
        if( b > c )
            max = b;
        else
            max = c;
    }
    printf("Najveći od tri broja %d, %d i %d je broj %d",
a, b, c, max);
}
```

## Dvostrana selekcija sa složenijim uvjetom

Što će se na zaslon ispisati sljedećim programom ?

```
#include <stdio.h>
int main()
{
    int    a = 30;
    double b = 2;
    long   c = 15;
    char   d = 'C';

    if (a + b > c || d < 'Z' && d > 'B' )
        printf("A");
    printf("B");
    if ( b + 10 > a - b * 2 && d < 'F' )
        printf("C");
    else
        printf("D");
}
```

Rješenje: ABD