

Programski jezik I

–Primjeri sa 2. termina predavanja –

Tipovi podataka i deklaracija promjenljivih, osnovni operatori

1. Šta će biti smješteno u promjenljivoj **rezultat** nakon izvršavanja sljedećeg koda:

```
01 | int a = 5, b = 3, c = 10, d = 4, e = 2;  
02 | int rezultat = a + b * 2 + (c - d % e) * (b - d) / 2;
```

Dopunite kod tako da ga možete provjeriti na računaru. Rezultat je 6.

2. Šta će biti smješteno u promjenljivoj **rezultat** nakon izvršavanja sljedećeg koda:

```
01 | int a = 5, b = 3, c = 10, d = 4, e = 2;  
02 | int rezultat = (a + b) * 2 + ((c - d % e) * (b - d)) / 2;
```

Dopunite kod tako da ga možete provjeriti na računaru. Rezultat je 11.

3. Šta će biti smješteno u varijablama **x**, **y** i **z** nakon izvršavanja sljedećih linija koda:

```
01 | int x = 5, y = 3, z = 7;  
02 | x += y;  
03 | y -= z;  
04 | z *= 2;  
05 | x /= 2;  
06 | y %= 4;  
07 | ++z;  
08 | --x;
```

Rezultat je: **x = 3, y = 0, z = 15;**

4. Šta će biti smješteno u varijablama **x**, **y** i **z** nakon izvršavanja sljedećih linija koda:

```
01 | int x = 5, y = 3, z = 7;  
02 | ++x;  
03 | x += y--;  
04 | y -= z++;  
05 | z *= 2;  
06 | x /= 2;  
07 | y %= 4;  
08 | ++z;  
09 | --x;
```

Rezultat je: **x = 3, y = -1, z = 17**

5. Odrediti sadržaj varijabli **r1** i **r2** nakon izvršavanja sljedećih linija koda:

```
01 | int a = 5, b = 3, c = 7;  
02 | int r1, r2;  
03 | r1 = (a > b) ? (a + b) : (a - b);  
04 | r2 = (b >= c) ? (b * c) : (b / c);
```

Rezultat je: `r1 = 8, r2 = 0`.

6. Odrediti sadržaj varijabli `j` i `k` nakon izvršavanja sljedećih linija koda:

```
01 | int x = 5, y = 3, z = 7;
02 | int j = (x > 3) && (y <= 5);
03 | int k = (x == 5) || (z != 7);
```

Rezultat je: `j = 1, k = 1`.

7. Odrediti sadržaj varijabli `r1` i `r2` nakon izvršavanja sljedećih linija koda:

```
01 | int p = 2, q = 3, r = 4, s = 5;
02 | int r1, r2;
03 | r1 = (p > q) ? (++r) : (--s);
04 | r2 = (r < s) ? (p * q) : (r % s);
```

Rezultat je: `r1 = 4, r2 = 0`. Za vježbu, ponoviti zadatak, ali tako da je treća linija koda izmijenjena u: `r1 = (p > q) ? (++r) : (s++);`

8. Odrediti sadržaj varijabli `i` i `j` nakon izvršavanja sljedećih linija koda:

```
01 | int i, j;
02 | j = 2;
03 | i = (j++, 3);
```

Rezultat je: `i = 3, j = 3`.

9. Odrediti sadržaj varijabli `r1` i `r2` nakon izvršavanja sljedećih linija koda:

```
01 | #include <stdio.h>
02 |
03 | int main() {
04 |     int x = 10, y = 3;
05 |     float z = 4.5;
06 |     float r1;
07 |     int r2;
08 |     r1 = (float)x / y;
09 |     r2 = (int)(z) + y;
10 | }
```

Rezultat je: `r1 = 3.33, r2 = 7`.

10. Deklarisati jednu cjelobrojnu varijablu, jednu `double` varijablu i jedan karakter. Zatim odštampati broj bajtova koji zauzima svaka od varijabli.

```
01 | #include <stdio.h>
02 |
03 | int main() {
04 |     int x;
05 |     double y;
06 |     char z;
07 |
08 |     printf("Velicina int: %lu bajtova\n", sizeof(x));
09 |     printf("Velicina double: %lu bajtova\n", sizeof(y));
10 |     printf("Velicina char: %lu bajtova\n", sizeof(z));
11 |
12 |     return 0; // napomenimo da ova linija koda ne utiče na rezultat
13 | }
```

Napomenimo da se `sizeof` vraća kao `size_t`, koji se obično štampa koristeći `%lu` formatni specifikator.

11. U nastavku slijede ilustracije za logičke operacije po bitovima. Voditi računa da je prikazana samo 8-bitna reprezentacija, iako su varijable zapravo sa većim brojem bita (što ne remeti smisao primjera).

(a) Negacija po bitovima (`~`):

```
01 | unsigned int x = 5;      // binarno: 0000 0101
02 | unsigned int y = ~x;    // binarno: 1111 1010
```

(b) Logička operacija I (`a & b`):

```
01 | unsigned int a = 5;      // binarno: 0000 0101
02 | unsigned int b = 3;      // binarno: 0000 0011
03 | unsigned int c = a & b;  // binarno: 0000 0001 (rezultat je 1)
```

(c) Logičko ILI (`a | b`):

```
01 | unsigned int a = 5;      // binarno: 0000 0101
02 | unsigned int b = 3;      // binarno: 0000 0011
03 | unsigned int c = a | b;  // binarno: 0000 0111 (rezultat je 7)
```

(d) Ekskluzivno ILI (`a ^ b`):

```
01 | unsigned int a = 5;      // binarno: 0000 0101
02 | unsigned int b = 3;      // binarno: 0000 0011
03 | unsigned int c = a ^ b;  // binarno: 0000 0110 (rezultat je 6)
```

(e) Operacije pomjeranja

```
01 | unsigned int x = 5;      // binarno: 0000 0101
02 | unsigned int y = x << 1; // binarno: 0000 1010 (rezultat je 10)
03 | unsigned int z = x >> 1; // binarno: 0000 0010 (rezultat je 2)
```