

7. Vježbe uz predavanja

1. Što će se ispisati sljedećim odsječkom programa:

```
char c, c1;
c = 'a' / 2*1.1;
printf("%d\n", c);

c1 = 1 + c++;
printf("%d %d\n", c, c1);
c1 = ++c + 12;
printf("%d %c %d %c\n", c, c, c1, c1);
```

2. Što će se ispisati sljedećim odsječkom programa:

```
int i1 = 5, i2, j1 = 5, j2, k1, k2;
i2 = ++i1 + 3;
printf("%d %d\n", i1, i2);
j2 = j1++ + 3;
printf("%d %d\n", j1, j2);
k1 = i2++ * --j2;
printf("%d %d %d\n", i2, j2, k1);
i2++;
++j2;
k2 = ++i2 * j2++;
printf("%d %d %d\n", i2, j2, k2);
```

3. Što će se ispisati sljedećim programskim odsječkom:

```
int i = 23, j = 13, k = 11, m;
printf("%d\n", i | j && k);
printf("%d\n", i | j & k);
m = i == j && k;
printf("%d %d\n", m, -!m < 0);
m = i ^ (j=13);
printf("%d %d\n", m, j);
j = 7;
m = 7;
m = i & ~(j==7);
printf("%d\n", m);
m = ~(~k | k);
printf("%d\n", m);
```

4. Što će se ispisati sljedećim programskim odsječkom:

```
int i = 6;
printf("%d\n", i << 3);
printf("%d\n", i);
```

5. Što će se ispisati sljedećim odsječkom programa:

```
int i = 7;
if (i = 3 & 4)
    printf("ISTINA JE %d\n", i);
else
    printf("NIJE ISTINA %d\n", i);
if (i = 3 && 4)
    printf("ISTINA JE %d\n", i);
else
    printf("NIJE ISTINA %d\n", i);
printf("Jos jednom upozoravam, NIJE ISTINA\n");
printf("Program završava\n");
```

6. Što će se ispisati obavljanjem sljedećeg programskog odsječka:

```
char x, y, z;
x = 2;
y = '1';
z = (y <= x) ? ( ++x ) : ( y += ++x );
printf("%d %c %d %c", x, y, y, z);
```

7. Što će se ispisati obavljanjem sljedećeg programskog odsječka:

```
int a = 3, b = 5, c = 9, d;
d = a + (b+=c);
printf("%d %d %d %d", a, b, c, d);
```

8. Izračunati i na zaslon ispisati sumu:

$$\frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{999} - \frac{1}{1000}$$

9. S tipkovnice učitavati i sumirati cijele brojeve dok god su oni djeljivi s 3 ili 7. Na kraju ispisati sumu učitanih brojeva. Podrazumijeva se da se zadnji broj koji je učitani (onaj zbog kojeg se prekida učitavanje) **ne dodaje** u sumu. Vodite računa o tome da već i prvi učitani broj može biti "neispravan".
10. S tipkovnice učitavati i sumirati cijele brojeve dok god su oni djeljivi s 3 ili 7. Na kraju ispisati sumu učitanih brojeva. Podrazumijeva se da se zadnji broj koji je učitani (onaj zbog kojeg se prekida učitavanje) **također dodaje** u sumu. Vodite računa o tome da već i prvi učitani broj može biti "neispravan".
11. S tipkovnice učitati cijeli broj n koji mora biti između 0 i 16 (uključivo s granicama). Ako broj nije ispravan, ispisati odgovarajuću poruku. Ako je broj ispravno upisan, učitati n binarnih znamenki i ispisati dekadski ekvivalent učitanih binarnih brojeva (ne primjenjuje se tehnika dvojnog komplementa, pa je dekadski ekvivalent sigurno pozitivan broj).

Npr., ako je korisnik upisao

```
4
1
1
0
1
```

program treba ispisati 13 (jer je $1101_2 = 13_{10}$).

Npr., ako je korisnik upisao
0

program treba ispisati 0.

12. Načinite program za izračunavanje n!. Vrijednost za n učitati s tipkovnice. Ako broj nije ispravan (mora biti cijeli broj veći ili jednak 0), ispisati odgovarajuću poruku.
13. Načinite program za pretvaranje nenegativnog (to znači pozitivnog ili nule) cijelog dekadskog broja u heksadekadski oblik (uzastopnim dijeljenjem sa 16). Nije potrebno provjeravati je li učitani broj nenegativan. Znamenke heksadekadskog broja smijete ispisati "obrnutim" redoslijedom. Npr. za uneseni dekadski broj 725, program treba ispisati 5D2. Za uneseni broj 0, program treba ispisati 0.
14. Isto kao prethodni zadatak, ali za učitani dekadski broj treba ispisati oktalni broj.
15. U varijablu tipa `int` s tipkovnice učitati nenegativni cijeli broj (ne treba kontrola je li broj nenegativan). Na zaslon ispisati binarni sadržaj registra te varijable (binarne znamenke se moraju ispisati ispravnim redoslijedom). Za izdvajanje bitova koristiti bitovni operator `&` i operator za posmak bitova prema desno. Može se pretpostaviti da se za pohranu `int` tipa podatka koristi 32 bita.

Uputa:

- ako se sadržaj registra `0100000000000000000000000000000010010` posmakne za **31** mjesta u desno, koja binarna znamenka originalnog broja će se pojaviti na krajnjoj desnoj poziciji?
- ako se sadržaj registra `0100000000000000000000000000000010010` posmakne za **30** mjesta u desno, koja binarna znamenka originalnog broja će se pojaviti na krajnjoj desnoj poziciji?
- ako se sadržaj registra `0100000000000000000000000000000010010` posmakne za **29** mjesta u desno, koja binarna znamenka originalnog broja će se pojaviti na krajnjoj desnoj poziciji? I tako dalje...
- vrijednost "krajnje desnog bita" varijable `x` može se dobiti operacijom `x & 1`

Rješenja svih zadataka provjeriti prevođenjem i testiranjem vlastitih programa!

Rješenja: NE GLEDATI prije nego sami pokušate riješiti zadatke

Rješenje 6. zadatka

Prvo se izračunava $y \leq x \rightarrow 0$

Budući je rezultat "laž", izraz (++x) se uopće **ne izračunava!**

Izračunava se samo (y+= ++x), a zatim se rezultat **tog** izraza pridružuje varijabli z.

Zbog prefiksnog oblika ++ operatora, varijabla x se **prvo** uvećava za 1, a zatim se njena vrijednost

dalje koristi u izrazu. Uvećana vrijednost varijable x je 3. $y += 3$ je isto što i $y = y + 3$

Konačno, x = 3, y = '4' (odnosno 52), z = '4' (odnosno 52)

Ispisat će se: 3 4 52 4

Rješenje 7. zadatka

Prvo se izračunava (b+=c) $\rightarrow b=14$, rezultat cijelog izraza (b+=c) je 14

d = a + 14, tj. d = 17

Ispisat će se:

3 14 9 17

Rješenje 8. zadatka

```
#include <stdio.h>
int main () {
    int i;
    float suma, brojnik;
    i = 1;
    suma = 0.f;
    while (i <= 1000) {
        brojnik = i % 2 ? 1.f : -1.f;
        suma += brojnik / i;
        i++;
    }
    printf("Suma je: %f\n", suma);
    return 0;
}
```

Rješenje 9. zadatka

```
#include <stdio.h>
int main () {
    int suma = 0, x;
    scanf("%d", &x);
    while (x % 3 == 0 || x % 7 == 0) {
        suma += x;
        scanf("%d", &x);
    }
    printf("Suma ucitanih brojeva je %d\n", suma);
    return 0;
}
```

Rješenje 10. zadatka

```
#include <stdio.h>
int main () {
    int suma = 0, x;
    scanf("%d", &x);
    suma += x;
    while (x % 3 == 0 || x % 7 == 0) {
        scanf("%d", &x);
        suma += x;
    }
    printf("Suma ucitanih brojeva je %d\n", suma);
    return 0;
}
```

Rješenje 11. zadatka

```
#include <stdio.h>
int main () {
    int n, znamenka, dekadski = 0;
    scanf("%d", &n);
    if (n < 0 || n > 16) {
        printf("Upisali ste neispravan broj\n");
    }
    else {
        while (n > 0) {
            scanf("%d", &znamenka);
            dekadski = dekadski*2 + znamenka;
            n--;
        }
        printf("%d\n", dekadski);
    }
    return 0;
}
```

Rješenje 12. zadatka

```
#include <stdio.h>
int main () {
    int n, fakt = 1;
    scanf("%d", &n);
    if (n < 0) {
        printf("Upisali ste neispravan broj\n");
    }
    else {
        while (n > 1) {
            fakt *= n;
            n--;
        }
        printf("%d\n", fakt);
    }
    return 0;
}
```

Rješenje 13. zadatka

```
#include <stdio.h>
int main () {
    int n, ostatak;
    scanf("%d", &n);
    if (n == 0) {
        printf("0\n");
    }
    else {
        while (n > 0) {
            ostatak = n % 16;
            if (ostatak < 10)
                printf("%d", ostatak);
            else
                printf("%c", 'A' + ostatak - 10);
            n = n / 16;
        }
    }
    return 0;
}
```

Rješenje 14. zadatka

```
#include <stdio.h>
int main () {
    int n, ostatak;
    scanf("%d", &n);
    if (n == 0)
        printf("0\n");
    else
        while (n > 0) {
            ostatak = n % 8;
            printf("%d", ostatak);
            n = n / 8;
        }
    return 0;
}
```

Rješenje 15. zadatka

```
#include <stdio.h>
int main () {
    int broj;
    int kolikoPosmaknutiDesno, pomocna;
    scanf("%d", &broj);
    printf("Upisali ste broj %d\n", broj);
    kolikoPosmaknutiDesno = 31;
    while (kolikoPosmaknutiDesno >= 0) {
        pomocna = broj >> kolikoPosmaknutiDesno;
        /* sada se u varijabli pomocna, na poziciji nultog bita (najmanje
           znacajnog bita) nalazi znamenka koja se u varijabli broj nalazi na poziciji
           kolikoPosmaknutiDesno. Vrijednost tog bita, 0 ili 1, moze se dobiti tako da se
           obavi operacija: pomocna & 000000000000000000000000000000012 */
        printf("%d", pomocna & 1);
        /* u sljedecem koraku posmaknuti za 30 mjesta, u sljedecem za 29 mjesta itd. */
        kolikoPosmaknutiDesno--;
    }
    printf("\n");
    return 0;
}
```