# 8. Dodatne vježbe

1. 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);</pre>
```

2. Što će se ispisati sljedećim programskim odsječkom:

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

3. Gdje se (i zašto) u sljedećem odsječku programa nalazi sintaktička pogreška:

```
float x, y;
int i;
x = 7.f;
y = 8.f;
y = x ^ 0xFF;
i = x |  'A' && y;
```

4. 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 zavrsava\n");
```

5. 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);
```

6. 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);
```

7. S tipkovnice učitati cijeli broj n koji mora biti između 0 i 16 (uključivo s granicama). Ako broj nije ispravan, ispisati odgovarajuću poruku. Nakon toga učitati n binarnih znamenki i ispisati dekadski ekvivalent učitanog binarnog broja (ne primjenjuje se tehnika dvojnog komplementa, pa je dekadski ekvivalent sigurno pozitivan broj).

Npr., ako je korisnik upisao

1 1

0

program treba ispisati 13.

Npr., ako je korisnik upisao

program treba ispisati 0.

Riješiti pomoću petlje s ispitivanjem uvjeta na početku.

- 8. Načinite program za izračunavanje n!. Vrijednost za n učitati s tipkovnice. Ako broj nije ispravan (mora biti cijeli broj veći ili jednaki 0), ispisati odgovarajuću poruku. Riješiti pomoću petlje s ispitivanjem uvjeta na početku.
- 9. Načinite program za pretvaranje nenegativnog cijelog dekadskog broja u heksadekadski oblik (uzastopnim dijeljenjem sa 16). 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. Riješiti pomoću petlje s ispitivanjem uvjeta na početku.
- 10. Isto kao prethodni zadatak, ali za učitani dekadski broj treba ispisati oktalni broj.
- 11. U varijablu tipa int 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.

# Rješenja: NE GLEDATI prije nego sami pokušate riješiti zadatke

# Rješenja zadataka 1-4 možete otkriti prevođenjem i izvođenjem vlastitih programa

```
Rješenje 5. zadatka
```

```
Prvo se izračunava y \le x \to 0 Budući je rezultat "laž", izraz ( ++x ) se uopće <u>ne izračunava!</u> 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 6. zadatka

```
Prvo se izračunava (b+=c) → b=14, rezultat cijelog izraza (b+=c) je 14 d = a + 14, tj. d = 17 Ispisat će se:

3 14 9 17
```

#### Rješenje 7. 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 8. 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 9. 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);
            printf("%c", 'A' + ostatak - 10);
         n = n / 16;
      }
   }
   return 0;
}
```

## Rješenje 10. 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 11. 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 bit koji se u varijabli broj nalazi na poziciji
       kolikoPosmaknutiDesno. Vrijednost tog bita, 0 ili 1, moze se dobiti tako da se
       obavi operacija: pomocna (and po bitovima) 0000000000000000000000000000012 */
     printf("%d", pomocna & 1);
     /* u sljedecem koraku posmaknuti za 30 mjesta, u sljedecem za 29 mjesta itd. */
     kolikoPosmaknutiDesno--;
  printf("\n");
  return 0;
```