

1. U varijablu tipa unsigned int s tipkovnice učitati nenegativni cijeli broj. 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 podatka tipa unsigned int koristi 32 bita.

Uputa:

- ako se sadržaj registra 1100100000000000000000000000000010010 posmakne za 31 mjesto u desno, koja binarna znamenka originalnog broja će se pojaviti na krajnjoj desnoj poziciji?
- ako se sadržaj registra 1100100000000000000000000000000010010 posmakne za 30 mjesta u desno, koja binarna znamenka originalnog broja će se pojaviti na krajnjoj desnoj poziciji?
- ako se sadržaj registra 1100100000000000000000000000000010010 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 $x \& 0x1$

Primjeri izvršavanja programa

```
Upisite nenegativni cijeli broj > 43.↵  
43(10) = 00000000000000000000000000000000101011(2)
```

```
Upisite nenegativni cijeli broj > 3355443218.↵  
3355443218(10) = 1100100000000000000000000000000010010(2)
```

2. U varijablu tipa unsigned int s tipkovnice učitati nenegativni cijeli broj. Na zaslon ispisati oktalni sadržaj registra te varijable (oktalne znamenke se moraju ispisati ispravnim redoslijedom). Za izdvajanje bitova (po tri binarne znamenke) koristiti bitovni operator & i operator za posmak bitova prema desno. Može se pretpostaviti da se za pohranu podatka tipa unsigned int koristi 32 bita.

Uputa:

- ako se sadržaj registra 1010110110000000000000000000000010010 posmakne za 30 mjesta u desno, koje tri binarne znamenke originalnog broja će se pojaviti na poziciji krajnja tri desna bita?
- ako se sadržaj registra 1010110110000000000000000000000010010 posmakne za 27 mjesta u desno, koje tri binarne znamenke originalnog broja će se pojaviti na poziciji krajnja tri desna bita?
- i tako dalje ...
- vrijednost "tri krajnja desna bita" varijable x može se dobiti operacijom $x \& 0x7$

Primjeri izvršavanja programa

```
Upisite nenegativni cijeli broj > 43.↵  
43(10) = 00000000053(8)
```

```
Upisite nenegativni cijeli broj > 3355443218.↵  
3355443218(10) = 31000000022(8)
```

3. U varijablu tipa unsigned int s tipkovnice učitati nenegativni cijeli broj. Na zaslon ispisati heksadekaski sadržaj registra te varijable (heksadekadske znamenke se moraju ispisati ispravnim redoslijedom). Za izdvajanje bitova (po četiri binarne znamenke) koristiti bitovni operator & i operator za posmak bitova prema desno. Može se pretpostaviti da se za pohranu podatka tipa unsigned int koristi 32 bita. Uputa: rješenje je vrlo slično rješenjima prethodna dva zadatka.

Primjeri izvršavanja programa

```
Upisite nenegativni cijeli broj > 43↵
43(10) = 0000002B(16)
```

```
Upisite nenegativni cijeli broj > 3355443218↵
3355443218(10) = C8000012(16)
```

4. Napisati program koji će s tipkovnice učitavati parove: vrijednost bita (0 ili 1) i poziciju bita (0-31) u varijabli broj kojeg treba postaviti na učitanoj vrijednosti bita. Ako je vrijednost bita i pozicija ispravno upisana, postaviti odgovarajući bit u varijabli broj (koja je na početku programa postavljena na vrijednost nula) te vrijednost varijable broj ispisati u dekadskom i binarnom obliku. Učitavanje parova bit-pozicija prekinuti kada se za vrijednost bita ili poziciju upiše neispravna vrijednost.

Primjer izvršavanja programa

```
Upisite bit i poziciju > 1 12↵
00000000000000000000000000000000(2) = 4096(10)
Upisite bit i poziciju > 1 10↵
00000000000000000000000000000000(2) = 5120(10)
Upisite bit i poziciju > 1 0↵
00000000000000000000000000000000(2) = 5121(10)
Upisite bit i poziciju > 0 10↵
00000000000000000000000000000000(2) = 4097(10)
Upisite bit i poziciju > 1 31↵
10000000000000000000000000000000(2) = 2147487745(10)
Upisite bit i poziciju > 1 30↵
11000000000000000000000000000000(2) = 3221229569(10)
Upisite bit i poziciju > 2 1↵
Pogresna vrijednost ili pozicija bita
```

5. Što će se ispisati sljedećim odsječkom programa (riješiti "na papiru" i zatim provjeriti izvršavanjem 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);
```

6. Što će se ispisati sljedećim odsječkom programa (riješiti "na papiru" i zatim provjeriti izvršavanjem 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);
```

7. Što će se ispisati sljedećim odsječkom programa (riješiti "na papiru" i zatim provjeriti izvršavanjem programa):

```
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);
```

8. Što će se ispisati sljedećim odsječkom programa (riješiti "na papiru" i zatim provjeriti izvršavanjem programa):

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

9. Što će se ispisati sljedećim odsječkom programa (riješiti "na papiru" i zatim provjeriti izvršavanjem programa):

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

10. Što će se ispisati sljedećim odsječkom programa (riješiti "na papiru" i zatim provjeriti izvršavanjem programa):

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

Rješenja:1. `#include <stdio.h>`

```
int main(void) {
    unsigned int broj;

    printf("Upisite nenegativni cijeli broj > ");
    scanf("%u", &broj);

    printf("%u(10) = ", broj);
    int posmakDesno;
    for (posmakDesno = 31; posmakDesno >= 0; posmakDesno--) {
        printf("%d", broj >> posmakDesno & 0x1);
    }
    printf("(2)");

    return 0;
}
```

2. `#include <stdio.h>`

```
int main(void) {
    unsigned int broj;

    printf("Upisite nenegativni cijeli broj > ");
    scanf("%u", &broj);

    printf("%u(10) = ", broj);
    int posmakDesno;
    for (posmakDesno = 30; posmakDesno >= 0; posmakDesno -= 3) {
        printf("%d", broj >> posmakDesno & 0x7);
    }
    printf("(8)");

    return 0;
}
```

3. #include <stdio.h>

```
int main(void) {
    unsigned int broj;

    printf("Upisite nenegativni cijeli broj > ");
    scanf("%u", &broj);

    printf("%u(10) = ", broj);
    int posmakDesno;
    for (posmakDesno = 28; posmakDesno >= 0; posmakDesno -= 4) {
        int znamenka = broj >> posmakDesno & 0xF;
        if (znamenka >= 0 && znamenka <= 9) {
            printf("%d", znamenka);
        } else {
            printf("%c", znamenka - 10 + 'A');
        }
    }
    printf("(16)");

    return 0;
}
```

4. #include <stdio.h>

```
int main(void) {
    unsigned int broj = 0u;
    int bit, poz;

    do {
        printf("Upisite bit i poziciju > ");
        scanf("%d %d", &bit, &poz);
        if ((bit == 0 || bit == 1) && poz >= 0 && poz <= 31) {
            if (bit == 1) {
                broj = broj | 0x1 << poz;
            } else {
                broj = broj & ~(0x1 << poz);
            }

            int posmakDesno;
            for (posmakDesno = 31; posmakDesno >= 0; posmakDesno--) {
                printf("%d", broj >> posmakDesno & 0x1);
            }
            printf("(2) = %u(10)\n", broj);
        } else {
            printf("Pogresna vrijednost ili pozicija bita");
        }
    } while ((bit == 0 || bit == 1) && poz >= 0 && poz <= 31);

    return 0;
}
```

5. -

6. -

7. -

8. -

9. Prvo se izračunava $y \leq x$ → rezultat je 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

10. Prvo se izračunava $(b += c)$ → rezultat je 14

Izračunava se $d = a + 14$ → rezultat je 17, pridružuje se varijabli d.

Ispisat će se: 3 14 9 17