

1. Dekadski broj 29 prikazati u obliku binarnog broja.
2. Binarni broj 10011011 prikazati u obliku dekadskog broja. Za prikaz ovog binarnog broja nije korištena tehnika dvojnog komplementa.
3. Registar od 8 bitova koristi se za prikaz brojeva tehnikom dvojnog komplementa. Koja dekadaska vrijednost je prikazana u registru, ako je sadržaj registra 00011011?
4. Registar od 8 bitova koristi se za prikaz brojeva tehnikom dvojnog komplementa. Koja dekadaska vrijednost je prikazana u registru, ako je sadržaj registra 10011011?
5. Dekadski broj -14 prikazati kao binarni broj u registru od 5 bitova, korištenjem tehnike dvojnog komplementa.
6. Dekadski broj -14 prikazati kao binarni broj u registru od 10 bitova, korištenjem tehnike dvojnog komplementa.
7. Koji se najveći i najmanji broj (izraziti u dekadskom obliku) može pohraniti u registru od 12 bita?
 - a) ako se ne koristi tehnika dvojnog komplementa (pohranjuju se samo pozitivni brojevi)
 - b) ako se koristi tehnika dvojnog komplementa (pohranjuju se i pozitivni i negativni brojevi)
8. Koliko najmanje bitova treba imati registar ako je u njega potrebno pohraniti dekadski broj 38?
 - a) ako se ne koristi tehnika dvojnog komplementa (pohranjuju se samo pozitivni brojevi)
 - b) ako se koristi tehnika dvojnog komplementa (pohranjuju se i pozitivni i negativni brojevi)
9. U binarnom brojevnom sustavu, uz primjenu tehnike dvojnog komplementa, koristeći registre veličine 5 bitova, obaviti operacije:
 - a) $4_{10} + 7_{10}$
 - b) $12_{10} - 5_{10}$
 - c) $7_{10} + 11_{10}$
 - d) $12_{10} - 16_{10}$

Rezultate provjeriti pretvorbom dobivenih binarnih rezultata u dekadске brojeve.

10. Dekadski broj 110 pretvoriti u oktalni broj:
 - a) direktno (uzastopnim dijeljenjem s 8)
 - b) indirektno, grupiranjem znamenaka binarnog broja
11. Dekadski broj 94 pretvoriti u heksadekadski broj:
 - a) direktno (uzastopnim dijeljenjem ss 16)
 - b) indirektno, grupiranjem znamenaka binarnog broja
12. Heksadekadske brojeve F2C i 4E napisati u obliku binarnih i oktalnih brojeva.
13. Oktalni broj 76431 napisati u obliku heksadekadskog broja.
14. Dekadski broj -9 pohraniti u registar od 5 bitova (tehnikom dvojnog komplementa). Rezultat prikazati kao:
 - a) binarni broj
 - b) heksadekadski broj
 - c) oktalni broj

15. Napisati program koji učitava cijele brojeve (int) dok god se ne upiše nula. Svaki prije toga upisani broj ispisati u obliku oktalnog i heksadekadskog broja. Uputa: koristiti konverzijske specifikacije koje cijeli broj mogu ispisati u oktalnom ili heksadekadskom obliku.

Primjer izvršavanja programa

```
Upisite dekadski cijeli broj > 2147483647↵
17777777777↵
7FFFFFFF↵
Upisite dekadski cijeli broj > -2147483648↵
20000000000↵
80000000↵
Upisite dekadski cijeli broj > 0↵
```

16. Napisati program koji učitava nenegativne cijele brojeve (unsigned int) u heksadekadskom obliku, dok god se ne upiše nula. Svaki prije toga upisani broj ispisati u obliku dekadskog broja. Uputa: koristiti konverzijsku specifikaciju koja nenegativni cijeli broj može učitati u heksadekadskom obliku.

Primjer izvršavanja programa

```
Upisite heksadekadski nenegativni cijeli broj > 1a↵
.....26↵
Upisite heksadekadski nenegativni cijeli broj > 7FFFFFFF↵
2147483647↵
Upisite heksadekadski nenegativni cijeli broj > FFFFFFFF↵
4294967295↵
Upisite heksadekadski nenegativni cijeli broj > 0↵
```

17. Napisati program kojim će se učitavati cijeli broj (int) dok god se upisuju brojevi koji odgovaraju ASCII kodovima velikih ili malih slova ili znamenki 0-9. Svaki ispravno upisani broj ispisati u obliku dekadskog broja i kao znak.

Primjer izvršavanja programa

```
Upisite cijeli broj > 65↵
65 A↵
Upisite cijeli broj > 97↵
97 a↵
Upisite cijeli broj > 49↵
49 1↵
Upisite cijeli broj > 47↵
```

18. Napisati program koji s tipkovnice učitava dva znaka te ispisuje sve znakove ASCII tablice koji se nalaze između ta dva znaka (uključujući).

Primjeri izvršavanja programa

```
Upisite dva znaka > dk↵
defghijk
```

```
Upisite dva znaka > Az↵
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz
```

19. Napisati program koji će ispisati tablicu kao u priloženom primjeru izvršavanja programa.
Napomena: očekuje se zadatak riješiti pomoću dvije programske petlje, a ne npr. ovako:

```
printf("A. a b c d e f .F\n");
printf("B. b c d e f g .G\n");
printf("C. c d e f g h .H\n");
printf("D. d e f g h i .I\n");
...
```

Primjer izvršavanja programa

```
A. a b c d e f .F↵
B. b c d e f g .G↵
C. c d e f g h .H↵
D. d e f g h i .I↵
E. e f g h i j .J↵
...
S. s t u v w x .X↵
T. t u v w x y .Y↵
U. u v w x y z .Z↵
```

20. U jednodimenzijско polje znakova učitati točno 10 znakova. Ispisati učitane članove polja, sortirati članove polja od manjih prema većim i ispisati novi, sortirani sadržaj polja. Upotreba pomoćnih polja nije dopuštena.

Primjeri izvršavanja programa

```
Upisite 10 znakova > Budimpesta↵
Original :Budimpesta↵
Sortirani:Badeimpstu
```

```
Upisite 10 znakova > New York↵
..↵
Original :New York↵
..↵
Sortirani:↵
.NYekorw
```

21. Napisati program kojim će se učitati prirodni broj n iz intervala $[1, 4250000000]$. Nije potrebno provjeravati je li upisan ispravan broj. Ispisati kvadrat učitano broj n .


Primjeri izvršavanja programa

```
Upisite prirodni broj u granicama [1, 4250000000] > 1↵  
1 * 1 = 1
```

```
Upisite prirodni broj u granicama [1, 4250000000] > 5↵  
5 * 5 = 25
```

```
Upisite prirodni broj u granicama [1, 4250000000] > 4250000000↵  
4250000000 * 4250000000 = 18062500000000000000
```

Rješenja:

- | | | | |
|----|-------------|-----------|---|
| 1. | 29 : 2 = 14 | ostatak 1 |  |
| | 14 : 2 = 7 | ostatak 0 | |
| | 7 : 2 = 3 | ostatak 1 | |
| | 3 : 2 = 1 | ostatak 1 | |
| | 1 : 2 = 0 | ostatak 1 | |

Broj 29 je pozitivan, stoga nije potrebno izračunavati dvojni komplement.

Rješenje: 11101

Provjera: $1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 29$

2. Prva znamenka jest jedinica, ali u zadatku piše da nije korištena tehnika dvojnog komplementa. To znači da se radi o pozitivnom broju:

$$1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 155$$

3. U zadatku piše da se za prikaz broja koristi tehnika dvojnog komplementa, ali prvi bit u registru nije jedinica. To znači da je u registru prikazan pozitivan broj. Vrijednost se određuje na isti način kao da se tehnika dvojnog komplementa uopće ne koristi:

$$1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 27$$

4. Za prikaz broja se koristi tehnika dvojnog komplementa, a prvi bit u registru jest jedinica. To znači da je u registru prikazan neki negativan broj x. Izračunavanjem dvojnog komplementa dobit će se broj koji je jednak po apsolutnoj vrijednosti, ali suprotnog predznaka (dakle, pozitivan broj):

$$\begin{array}{rcl}
 10011011 & \rightarrow & x \\
 01100100 & \rightarrow & \text{jedinični komplement} \\
 + \quad 1 & & \text{dodaje se jedan kako bi se dobio dvojni komplement} \\
 \hline
 = 01100101 & \rightarrow & -x
 \end{array}$$

Dobiveni broj -x je pozitivan broj (prvi bit mu nije jedinica), stoga se lako može odrediti o kojem se dekadskom broju radi:

$$1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 101_{10}$$

Ako je $-x=101_{10}$, tada je $x=-101_{10}$

Konačno rješenje jest: u registru je pohranjen dekadski broj -101_{10}

5. Negativni cijeli brojevi prikazuju se tehnikom dvojnog komplementa. Nije moguće direktno odrediti binarni prikaz broja -14, stoga se prvo određuje binarni prikaz pozitivnog broja 14 (uzastopnim dijeljenjem s 2). Voditi računa o tome da registar ima 5 bitova!

$$\begin{array}{rcl}
 +14_{10} & = & 01110_2 \\
 01110 & \rightarrow & +14_{10} \\
 10001 & \rightarrow & \text{jedinični komplement} \\
 + \quad 1 & \rightarrow & \text{dodaje se jedinica kako bi se dobio dvojni komplement} \\
 \hline
 = 10010 & \rightarrow & \text{dvojni komplement}
 \end{array}$$

Konačno rješenje: -14 prikazan u tehnici dvojnog komplementa u registru od 5 bitova jest 10010

6. Slično kao u prethodnom zadatku. Treba voditi računa da se sada radi o 10-bitnom registru!

$$\begin{array}{rcl}
 +14_{10} & = & 0000001110_2 \\
 0000001110 & \rightarrow & +14_{10} \\
 1111110001 & \rightarrow & \text{jedinični komplement} \\
 + & 1 & \rightarrow \text{dodaje se jedinica kako bi se dobio dvojni komplement} \\
 = & 1111110010 & \rightarrow \text{dvojni komplement}
 \end{array}$$

-14 prikazan u tehnici dvojnog komplementa u registru od 10 bitova jest 1111110010

7. a) Ako se ne koristi tehnika dvojnog komplementa, raspon brojeva koji se može prikazati u registru od n bitova jest $[0, 2^n-1]$. Najveći broj koji se može prikazati u 12-bitnom registru jest $2^{12}-1 = 4095$. Najmanji broj koji se može prikazati jest 0.
- b) Ako se koristi tehnika dvojnog komplementa, raspon brojeva koji se može prikazati u registru od n bitova jest $[-(2^{n-1}), 2^{n-1}-1]$. Najveći broj koji se može prikazati u 12-bitnom registru jest $2^{11}-1 = 2047$. Najmanji broj koji se može prikazati jest -2048.
8. a) Ako se ne koristi tehnika dvojnog komplementa, raspon brojeva koji se može prikazati u registru od 5 bitova jest $[0, 31]$, a u registru od 6 bitova $[0, 63]$. Potreban je registar od 6 bitova.
- b) Ako se koristi tehnika dvojnog komplementa, raspon brojeva koji se može prikazati u registru od 6 bitova jest $[-32, 31]$, a u registru od 7 bitova $[-64, 63]$. Potreban je registar od 7 bitova.

9. a)

$$\begin{array}{rcl}
 00100 & \rightarrow & 4 \\
 + 00111 & \rightarrow & 7 \\
 = 01011 & \rightarrow & 11
 \end{array}$$

b)

$$\begin{array}{rcl}
 01100 & \rightarrow & 12 \\
 + 11011 & \rightarrow & -5 \\
 = 00111 & \rightarrow & 7
 \end{array}$$

c)

$$\begin{array}{rcl}
 00111 & \rightarrow & 7 \\
 + 01011 & \rightarrow & 11 \\
 = 10010 & \rightarrow & -14
 \end{array}$$

U ovom slučaju rezultat nije kakav bi se očekivao jer se u registru od 5 bitova, u tehnici dvojnog komplementa, broj 18 ne može prikazati.

d)

$$\begin{array}{rcl}
 01100 & \rightarrow & 12 \\
 + 10000 & \rightarrow & -16 \\
 = 11100 & \rightarrow & -4
 \end{array}$$

10. a)

$$\begin{array}{rcl}
 110 : 8 = 13 & \text{ostatak } 6 & \uparrow \\
 13 : 8 = 1 & \text{ostatak } 5 & \\
 1 : 8 = 0 & \text{ostatak } 1 &
 \end{array}$$

Rješenje: 156_8

Provjera: $1 \cdot 8^2 + 5 \cdot 8^1 + 6 \cdot 8^0 = 110_{10}$

$$b) \quad 110_{10} = 1101110_2$$

Binarne znamenke grupirati po tri. PAZITI da se grupiranje obavi "s desna na lijevo":

$$1 \ 101 \ 110_2 = 156_8$$

$$11. \ a) \quad \begin{array}{ll} 94 : 16 = 5 & \text{ostatak } 14 \\ 5 : 16 = 0 & \text{ostatak } 5 \end{array} \quad \uparrow$$

Rješenje: $5E_{16}$

$$\text{Provjera: } 5 \cdot 16^1 + 14 \cdot 16^0 = 94_{10}$$

$$b) \quad 94_{10} = 1011110_2$$

Binarne znamenke grupirati po četiri. PAZITI da se grupiranje znamenki obavi "s desna na lijevo":

$$101 \ 1110_2 = 5E_{16}$$

12. Svaka heksadekadaska znamenka pretvara se u četiri binarne:

$$F \ 2 \ C_{16} = 1111 \ 0010 \ 1100_2$$

$$4 \ E_{16} = 0100 \ 1110_2$$

Heksadekadski broj se lako pretvara u oktalni: heksadekadski broj treba napisati kao binarni broj, zatim binarne znamenke grupirati u grupe po tri. PAZITI da se grupiranje znamenki obavi "s desna na lijevo":

$$F \ 2 \ C_{16} = 1111 \ 0010 \ 1100_2 = 111 \ 100 \ 101 \ 100_2 = 7454_8$$

$$4 \ E_{16} = 0100 \ 1110_2 = 01 \ 001 \ 110_2 = 116_8$$

13. Oktalni broj se lako pretvara u heksadekadski: oktalni broj treba napisati kao binarni broj, zatim binarne znamenke grupirati u grupe po četiri. PAZITI da se grupiranje znamenki obavi "s desna na lijevo":

$$7 \ 6 \ 4 \ 3 \ 1_8 = 111 \ 110 \ 100 \ 011 \ 001_2 = 111 \ 1101 \ 0001 \ 1001_2 = 7D19_{16}$$

$$14. \ -9_{10} = 10111_2 = 17_{16} = 27_8$$

15. #include <stdio.h>

```
int main(void) {
    int n;

    do {
        printf("Upisite dekadski cijeli broj > ");
        scanf("%d", &n);
        if (n != 0) {
            printf("%11o\n%8X\n", n, n);
        }
    } while (n != 0);

    return 0;
}
```

16. #include <stdio.h>

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

    do {
        printf("Upisite heksadekadski nenegativni cijeli broj > ");
        scanf("%x", &n);
        if (n != 0) {
            printf("%11u\n", n);
        }
    } while (n != 0);

    return 0;
}
```

17. #include <stdio.h>

```
int main(void) {
    int n;

    do {
        printf("Upisite cijeli broj > ");
        scanf("%d", &n);
        if ((n >= 'A' && n <= 'Z') ||
            (n >= 'a' && n <= 'z') ||
            (n >= '0' && n <= '9')) {
            printf("%d %c\n", n, n);
        }
    } while ((n >= 'A' && n <= 'Z') ||
            (n >= 'a' && n <= 'z') ||
            (n >= '0' && n <= '9'));

    return 0;
}
```

18. #include <stdio.h>

```
int main(void) {
    char c1, c2, c;

    printf("Upisite dva znaka > ");
    scanf("%c%c", &c1, &c2);
    for (c = c1; c <= c2; c = c + 1) {
        printf("%c", c);
    }

    return 0;
}
```


19. #include <stdio.h>

```
int main(void) {
    char i, j;

    for (i = 'A'; i <= 'U'; i = i + 1) {
        printf("%c. ", i);
        for (j = i + 'a' - 'A'; j < i + 'a' - 'A' + 6; j = j + 1) {
            printf("%c ", j);
        }
        printf(" %.5c\n", i + 5);
    }

    return 0;
}
```

20. #include <stdio.h>

```
#define MAKS 10

int main(void) {
    char polje[MAKS];
    printf("Upisite 10 znakova > ");

    int i;
    for (i = 0; i < MAKS; i = i + 1) {
        scanf("%c", &polje[i]);
    }

    printf("Original :");
    for (i = 0; i < MAKS; i = i + 1) {
        printf("%c", polje[i]);
    }

    int j, ind_min;
    for (i = 0; i < MAKS - 1; i = i + 1) {
        ind_min = i + 1;
        for (j = i + 2; j < MAKS; j = j + 1) {
            if (polje[j] < polje[ind_min]) {
                ind_min = j;
            }
        }

        if (polje[ind_min] < polje[i]) {
            char pomocna = polje[i];
            polje[i] = polje[ind_min];
            polje[ind_min] = pomocna;
        }
    }

    printf("\nSortirani:");
    for (i = 0; i < MAKS; i = i + 1) {
        printf("%c", polje[i]);
    }

    return 0;
}
```

```
21. #include <stdio.h>
```

```
int main(void) {  
    unsigned int n;  
    unsigned long long kvadrat;  
  
    printf("Upisite prirodni broj u granicama [1, 4250000000] > ");  
    scanf("%u", &n);  
  
    kvadrat = 1ULL * n * n;  
    printf("%u * %u = %llu", n, n, kvadrat);  
  
    return 0;  
}
```