

**Napomene:**

- Savjetuje se navedene zadatke riješiti ubrzo nakon predavanja
- Savjetuje se ne gledati rješenja prije nego se pokuša samostalno riješiti zadatke

## 9. vježbe uz predavanja

1. S tipkovnice u char varijablu `c` učitajte jedan znak. Ako je učitano veliko slovo, ispisati poruku "To je veliko slovo". Ako je učitano malo slovo, ispisati poruku "To je malo slovo". Ako je učitana znamenka (0-9), ispisati poruku "To je znamenka". Ako nije učitano ni slovo ni znamenka, ispisati poruku "To je neki drugi znak".
2. S tipkovnice učitati koeficijente kvadratne jednadžbe  $ax^2 + bx + c = 0$ . Zavisno od učitanih koeficijenata, rješenja (korijene) kvadratne jednadžbe ispisati u jednom od tri navedena oblika:

R = xxxxx.xxxx

ili

R1 = xxxxx.xxxx

R2 = xxxxx.xxxx

ili

C1 = xxxxx.xxxx + xxxxx.xxxx \* i

C2 = xxxxx.xxxx - xxxxx.xxxx \* i

3. S tipkovnice u char varijablu `z1` učitajte znak. Ako učitani znak nije heksadekadska znamenka (0-9, A-F, a-f), ispisati poruku "Nije dobra znamenka" i završiti program. Inače, ispisati dekadski ekvivalent učitane heksadekadske znamenke. Npr. ako je učitani znak `e`, treba ispisati 14.
4. **Riješiti pomoću petlje s ispitivanjem uvjeta na početku.** Izračunati i na zaslon ispisati sumu:

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

5. **Riješiti pomoću petlje s ispitivanjem uvjeta na početku.** 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".
6. **Riješiti pomoću petlje s ispitivanjem uvjeta na početku.** 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".

7. **Riješiti pomoću petlje s ispitivanjem uvjeta na početku.** 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 učitano binarnog broja (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.

8. **Riješiti pomoću petlje s ispitivanjem uvjeta na početku.** 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.
9. **Riješiti pomoću petlje s ispitivanjem uvjeta na početku.** 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.
10. **Riješiti pomoću petlje s ispitivanjem uvjeta na početku.** Isto kao prethodni zadatak, ali za učitani dekadski broj treba ispisati oktalni broj.
11. **Riješiti pomoću petlje s ispitivanjem uvjeta na početku.** 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  $x \& 1$

12. Riješiti pomoću petlje s ispitivanjem uvjeta na kraju (*do-while*). Usporediti s rješenjem u kojem se koristila petlja s ispitivanjem uvjeta na početku. Izračunati i na zaslon ispisati sumu:

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

13. Riješiti pomoću petlje s ispitivanjem uvjeta na kraju (*do-while*). Usporediti s rješenjem u kojem se koristila petlja s ispitivanjem uvjeta na početku. 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 učitao (onaj zbog kojeg se prekida učitavanje) **ne dodaje** u sumu. Vodite računa o tome da već i prvi učitani broj može biti "neispravan".

14. Riješiti pomoću petlje s ispitivanjem uvjeta na kraju (*do-while*). Usporediti s rješenjem u kojem se koristila petlja s ispitivanjem uvjeta na početku. 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 učitao (onaj zbog kojeg se prekida učitavanje) **također dodaje** u sumu. Vodite računa o tome da već i prvi učitani broj može biti "neispravan".

15. Napisati dva rješenja. Jedno rješenje pomoću petlje s ispitivanjem uvjeta na početku i jedno rješenje pomoću petlje s ispitivanjem uvjeta na kraju. Usporediti rješenja. S tipkovnice učitati nenegativan cijeli broj N (N može biti nula!). Na zaslon ispisati N kvocijenata (može biti niti jedan) u obliku realnih brojeva ukupne širine 7 znakova, s pet decimala iza decimalne točke):

$$\frac{1}{2} \quad \frac{2}{3} \quad \frac{3}{4} \quad \dots \quad \frac{N}{N+1}$$

16. S tipkovnice učitavati cijele brojeve dok god ne bude upisan broj manji ili jednak nuli. Na zaslon ispisati sumu učitanih pozitivnih brojeva podijeljenu s najvećim do tada učitanim pozitivnim brojem i pomnoženu s najmanjim do tada učitanim pozitivnim brojem. Ako nije upisan niti jedan pozitivan broj, ispisati odgovarajuću poruku. Primjer: ako korisnik utipka brojeve 2 5 3 2 2 -22, program treba ispisati 5.600000

17. Napišite program koji s tipkovnice učitava dva znaka te ispisuje sve znakove ASCII tablice koji se nalaze između ta dva znaka. Npr., ako se učitaju znakovi d i k, program ispisuje defghi jk.

18. Načinite program koji će ispisati sljedeću tablicu:

```
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
... itd.
S. s t u v w x .X
T. t u v w x y .Y
U. u v w x y z .Z
```

Očekuje se da zadatak riješite 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");
... itd.
```

19. Načinite program kojim će se učitati član niza  $a_1$  koji mora biti cijeli broj veći od 1. Nije potrebno provjeravati ispravnost unesenog broja. Ispisati članove niza  $a_1, a_2, a_3, \dots, a_n$  i ukupan broj članova niza. Član niza  $a_i$  (za  $i > 1$ ) izračunava se na sljedeći način:

ako je  $a_{i-1}$  paran, tada je  $a_i = a_{i-1} / 2$   
ako je  $a_{i-1}$  neparan, tada je  $a_i = 3 a_{i-1} + 1$

Niz završava članom  $a_n = 2$

## Rješenja

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

    int main (void) {
        char c;
        scanf("%c", &c);
        if (c >= 'A' && c <= 'Z')
            printf("To je veliko slovo\n");
        else if (c >= 'a' && c <= 'z')
            printf("To je malo slovo\n");
        else if (c >= '0' && c <= '9')
            printf("To je znamenka\n");
        else
            printf("To je neki drugi znak\n");

        return 0;
    }
```

Uočite da je sljedeće rješenje loše (objasnite zašto), iako "program radi":

### Loše rješenje !!!

```
#include <stdio.h>

int main (void) {
    char c;
    scanf("%c", &c);
    if (c >= 'A' && c <= 'Z')
        printf("To je veliko slovo\n");
    if (c >= 'a' && c <= 'z')
        printf("To je malo slovo\n");
    if (c >= '0' && c <= '9')
        printf("To je znamenka\n");
    if (! (c >= 'A' && c <= 'Z' ||
          c >= 'a' && c <= 'z' ||
          c >= '0' && c <= '9'))
        printf("To je neki drugi znak\n");

    return 0;
}
```

2.

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

int main (void) {
    double a, b, c;
    double q, diskrim;
    double r;
    double r1, r2;
    double cReal, cImag;

    printf("Upisite koeficijente kvadratne jednadzbe a, b, c: ");
    scanf("%lf %lf %lf", &a, &b, &c);

    diskrim = b * b - 4. * a * c; /* diskriminanta */
    if (diskrim == 0) {
        /* postoji samo jedno rjesenje */
        r = -b / ( 2. * a );
        printf ("R = %10.4f\n", r);
    } else if (diskrim > 0) {
        /* Rjesenja su realna */
        q = sqrt (diskrim); /* ili pow (diskrim, 1./2) */
        r1 = ( -b + q ) / ( 2. * a );
        r2 = ( -b - q ) / ( 2. * a );
        printf ("R1 = %10.4f\n"
                "R2 = %10.4f\n"
                , r1
                , r2);
    } else {
        /* Rjesenja su konjugirano kompleksni broj */
        q = sqrt (-diskrim); /* ili pow (-diskrim, 1./2) */
        cReal = -b / (2. * a ) ;
        cImag = q / ( 2. * a );
        printf ("C1 = %10.4f + %10.4f * i\n"
                "C2 = %10.4f - %10.4f * i\n"
                , cReal, cImag
                , cReal, cImag);
    }
    return 0;
}
```

3. `#include <stdio.h>`

```
int main (void) {
    char z1;
    char dekadskiBroj;
    scanf("%c", &z1);
    if (z1 >= 'A' && z1 <= 'F' ||
        z1 >= 'a' && z1 <= 'f' ||
        z1 >= '0' && z1 <= '9' ) {
        /* dobar znak */
        if (z1 >= 'A' && z1 <= 'F')
            dekadskiBroj = z1 - 65 + 10;
        else if (z1 >= 'a' && z1 <= 'f')
            dekadskiBroj = z1 - 97 + 10;
        else
            dekadskiBroj = z1 - 48;
        printf("heksadekadski %c = dekadski %d\n", z1, dekadskiBroj);
    }
    else {
        printf("Nije dobra znamenka\n");
    }
    return 0;
}
```

ili

`#include <stdio.h>`

```
int main (void) {
    char z1;
    scanf("%c", &z1);
    if (z1 >= 'A' && z1 <= 'F')
        printf("heksadekadski %c = dekadski %d\n", z1, z1 - 65 + 10);
    else if (z1 >= 'a' && z1 <= 'f')
        printf("heksadekadski %c = dekadski %d\n", z1, z1 - 97 + 10);
    else if (z1 >= '0' && z1 <= '9')
        printf("heksadekadski %c = dekadski %d\n", z1, z1 - 48);
    else
        printf("Nije dobra znamenka\n");
    return 0;
}
```

4. `#include <stdio.h>`

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

5. `#include <stdio.h>`

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

6. `#include <stdio.h>`

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

7. `#include <stdio.h>`

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

8. `#include <stdio.h>`

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



9. `#include <stdio.h>`

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

10. `#include <stdio.h>`

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

11. `#include <stdio.h>`

```
int main (void) {
    unsigned nt broj;
    int kolikoPosmaknutiDesno, pomocna;
    scanf("%u", &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;
}
```

## Rješenje 12. zadatka

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

s petljom s ispitivanjem uvjeta na početku

```
#include <stdio.h>
int main (void) {
    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;
}
```

Tijelo petlje će se sigurno izvršiti točno tisuću puta. Jednako je prikladno koristiti obje vrste petlji. Napomena: u ovom zadatku bi najprikladnija vrsta petlje bila petlja s poznatim brojem ponavljanja, koja za sada još nije obrađena na predavanjima i zato se ne koristi u rješenju.

## Rješenje 13. zadatka

```
#include <stdio.h>
int main (void) {
    int suma = 0, x;
    do {
        scanf("%d", &x);
        if (x % 3 == 0 || x % 7 == 0)
            suma += x;
    } while (x % 3 == 0 || x % 7 == 0);

    printf("Suma ucitanih brojeva je %d\n",
        suma);
    return 0;
}
```

s petljom s ispitivanjem uvjeta na početku

```
#include <stdio.h>
int main (void) {
    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;
}
```

Jedan dio tijela petlje se mora izvršiti barem jednom (scanf), a drugi dio tijela petlje (sumiranje) se možda neće trebati izvršiti niti jednom. Jednako je (ne)prikladno koristiti obje vrste petlji.

## Rješenje 14. zadatka

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

s petljom s ispitivanjem uvjeta na početku

```
#include <stdio.h>
int main (void) {
    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;
}
```

Cijelo tijelo petlje se mora izvršiti barem jednom. U ovom zadatku očito je prikladnije koristiti petlju s ispitivanjem uvjeta na kraju.

## Rješenje 15. zadatka

```
#include <stdio.h>
int main (void) {
    int n, i = 1;
    scanf("%d", &n);
    while (i <= n) {
        printf("%7.5f\n", (float)i/(i+1));
        ++i;
    }
    return 0;
}

#include <stdio.h>
int main (void) {
    int n, i = 1;
    scanf("%d", &n);
    do {
        if (i <= n) {
            printf("%7.5f\n", (float)i/(i+1));
            ++i;
        }
    } while (i <= n);
    return 0;
}
```

Moguće je da tijelo petlje neće trebati izvršiti niti jednom. U ovom zadatku očito je prikladnije koristiti petlju s ispitivanjem uvjeta na početku.

Napomena: u ovom zadatku bi najprikladnija vrsta petlje bila petlja s poznatim brojem ponavljanja, koja za sada još nije obrađena na predavanjima i zato se ne koristi u rješenju.

## Rješenje 16. zadatka

```
#include <stdio.h>

int main (void) {
    int suma = 0, najmanji, najveći, ucitani;
    do {
        scanf("%d", &ucitani);
        if (ucitani > 0) {
            if (suma == 0)
                najmanji = najveći = ucitani;
            suma += ucitani;
            if (ucitani < najmanji)
                najmanji = ucitani;
            if (ucitani > najveći)
                najveći = ucitani;
        }
    } while (ucitani > 0);
    if (suma > 0)
        printf("Rezultat je: %f\n", (float)suma / najveći * najmanji);
    else
        printf("Nije ucitan niti jedan pozitivan broj\n");
    return 0;
}
```

## Rješenje 17. zadatka

```
#include <stdio.h>

int main(void) {
    char c1, c2, c;
    scanf("%c %c", &c1, &c2);
    c = c1;
    while (c <= c2)
        printf("%c", c++);

    printf("\n");
    return 0;
}
```

**Komentirajte:** što će se dogoditi ako se umjesto znakova d i k s tipkovnice učitaju znakovi k i d ?  
Što će se dogoditi ako se umjesto znakova d i k s tipkovnice učitaju znakovi d i d ?

## Rješenje 18. zadatka

```
#include <stdio.h>

int main(void) {
    char i = 'A', j;
    while (i <= 'U') {
        printf("%c. ", i);
        j = i + 'a' - 'A';
        while (j < i + 'a' - 'A' + 6) {
            printf("%c ", j++);
        }
        printf(".*c\n", i++ + 5);
    }
    return 0;
}
```

Vanjska petlja mijenja vrijednost varijable i od 'A' do 'U'. Na početku svakog retka se ispisuje vrijednost varijable i (u formatu %c), a na kraju retka ispisuje se slovo koje se u ASCII tablici nalazi "5 mjesta dalje" od slova koje se ispisalo na početku retka, te na samom kraju znak za skok u novi red.

Unutarnja petlja mijenja vrijednost varijable j od "male verzije" slova koje je ispisano na početku retka, do slova koje se u ASCII tablici nalazi "6 mjesta dalje" od početnog malog slova.

## Rješenje 19. zadatka

```
#include <stdio.h>
int main(void) {
    int ai, brClanova = 0;
    printf("Upisite cijeli broj veci od 1: ");
    scanf("%d", &ai);
    while (ai >= 2) {
        brClanova = brClanova + 1;
        printf ("Clan a%d = %d\n", brClanova, ai);
        /* izracunaj sljedeci clan */
        if (ai % 2 != 0) /* prethodni clan je neparan */
            ai = 3 * ai + 1;
        else /* prethodni clan je paran */
            ai = ai / 2;
    }
    printf ("Ukupno %d clanova.\n", brClanova);
    return 0;
}
```

Komentirajte:

- što se dešava nakon što se obavi naredba kojom se ispisao član niza 2?
- što će se desiti ako korisnik kao prvi član niza unese broj  $\leq 1$ ?