

Napomene:

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

8. Vježbe uz predavanja

1. Izračunati i na zaslon ispisati sumu:

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

2. 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".
3. 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".
4. 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.

5. 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.

11. 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 učitani (onaj zbog kojeg se prekida učitavanje) **također dodaje** u sumu. Vodite računa o tome da već i prvi učitani broj može biti "neispravan".
12. 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 pozitivan cijeli broj N. Ukoliko je broj ispravan, na zaslon ispisati N kvocijenata (u obliku realnih brojeva):

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

13. 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. Ukoliko 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
14. 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 defghijk.
15. 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.
```

Rješenja svih zadataka provjeriti prevođenjem i testiranjem vlastitih programa!

Rješenja:

Rješenje 1. 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 2. 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 3. 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 4. 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 5. 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 6. 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;
        }
    }
}
```



```
    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 10. zadatka

```
#include <stdio.h>  
int main () {  
    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 () {  
    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 11. zadatka

```
#include <stdio.h>  
int main () {  
    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 () {  
    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 12. zadatka

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

```
#include <stdio.h>
int main () {
    int n, i = 1;
    scanf("%d", &n);
    do {
        if (i <= n) {
            printf("%f\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 13. zadatka

```
#include <stdio.h>

int main () {
    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 14. zadatka

```
#include <stdio.h>

int main() {
    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 15. zadatka

```
#include <stdio.h>

int main() {
    char i = 'A', j;
    while (i <= 'U') {
        printf("%c. ", i);
        j = i + 'a' - 'A';
        while (j < i + 'a' - 'A' + 6) {
            printf("%c ", j++);
        }
        printf(".\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.