

## 9. Dodatne vježbe

1. Učitati cijeli broj  $n$  koji mora biti između 0 i 32 (uključivo s granicama). Nakon toga 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. Riješiti pomoću petlje s ispitivanjem uvjeta na početku.

2. Prethodni zadatak riješiti pomoću petlje s unaprijed poznatim brojem ponavljanja. Koja je od dvije vrste petlji primjerenija za rješavanje ovog zadatka?
3. Načinite program za izračunavanje  $n!$ . Vrijednost za  $n$  učitati s tipkovnice uz kontrolu je li vrijednost ispravno zadana (mora biti cijeli broj veći ili jednaki 0). Načinite tri rješenja: s petljom s unaprijed poznatim brojem ponavljanja; s petljom s ispitivanjem uvjeta na početku; s petljom s ispitivanjem uvjeta na kraju. Koju vrstu petlje treba odabrati za rješavanje ovog zadatka?
4. Načinite program za "pogađanje broja  $\pi$ ". Korisnik, koji ne zna vrijednost broja  $\pi$ , uzastopnim upisivanjem brojeva pokušava pogoditi o kojoj se vrijednosti radi. Program mu odgovara:

```
Upisao si premalen broj.
Upisao si prevelik broj.
Nakon x koraka upisao si dovoljno točan broj.
```

Posljednja poruka se ispisuje, te se program prekida onda kada korisnik upiše vrijednost koja je točna barem na treću decimalu (3 točne znamenke iza decimalne točke), npr. ako korisnik upiše 3.14196. Na kraju dokumenta napisano je rješenje, ali nemojte ga gledati dok sami ne pokušate riješiti zadatak.

5. 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.
6. Načinite program koji će s tipkovnice učitati nenegativni cijeli broj iz intervala  $[0, 4294967295]$ . Učitani broj treba ispisati u oktalnom obliku. Npr. za učitani broj 250 treba ispisati 00000000372; za učitani broj 4294967295 treba ispisati 3777777777. Zadatak riješite tako da grupe od po tri bita pretvarate u oktalne znamenke. Za određivanje grupa po tri bita koristite operator posmaka u desno (za tri mjesta) i bitovni operator `&`. Na kraju ovog dokumenta napisano je rješenje, ali nemojte ga gledati dok sami ne pokušate riješiti zadatak. **Uputa:** vrijednost tipa podatka `unsigned` može se učitati po formatu `%u`.
7. Isto kao prethodni zadatak, ali učitani dekadski broj treba pretvoriti u heksadekadski.

**Ne gledajte ova rješenja dok sami ne pokušate riješiti zadatke!**

## Rješenje 4. zadatka

```
#include <stdio.h>

#define PI 3.141

int main() {
    float broj;
    int brKoraka = 0;
    int decZnamenke;
    do {
        brKoraka++;
        printf("Unesite broj: ");
        scanf("%f", &broj);
        decZnamenke = (broj - (int)broj) * 1000;
        if (decZnamenke != 141)
            if (broj > PI)
                printf("Upisao(la) si preveliki broj\n");
            else
                printf("Upisao(la) si premalen broj\n");
    } while (decZnamenke != 141);
    printf("Nakon %d koraka upisao(la) si dovoljno tocan broj\n", brKoraka);

    return 0;
}
```

Prve tri znamenke iza decimalne točke dobivene su tako da se od broja oduzeo njegov cijeli dio, dobiveni rezultat se pomnožio s 1000, te se rezultat pridružio u int varijablu, čime su odbačene sve preostale decimale. Npr. 3.1456 → 0.1456 → 145.6 → 145

Testiranjem programa, u nekim se slučajevima može dobiti na prvi pogled neočekivan rezultat. Npr. ako korisnik preko tipkovnice upiše broj 3.142, broj će se prepoznati kao dovoljno točan! Problem je u tome što korisnik zapravo nije u varijablu broj upisao 3.142, nego 3.141999996 (to je zato jer se 3.142 ne može prikazati s konačnim brojem binarnih znamenaka). To možete provjeriti tako da broj nakon učitavanja ispišete prema formatu %12.9f

## Rješenje 7. zadatka

```
#include <stdio.h>

int main() {
    unsigned int a;
    int i;
    printf("Upisite cijeli broj a: ");
    scanf ("%u", &a);
    for (i = 10; i >= 0; i--) {
        printf("%d", a >> 3*i & 0x7);
    }
    printf ("\n");
    return 0;
}
```