

## PETLJE

Ponekad se dijelovi programa trebaju ponoviti više puta (iterirati). Struktura koja to omogućava naziva se programska petlja. Ponavljati se može: unaprijed zadani broj puta (konačne petlje), sve dok je ispunjen određeni uslov (uslovne petlje).

Struktura petlje se može ostvariti naredbama:

- for,
- while,
- do – while.

## FOR PETLJA

for petlja se najčešće koristi ako se dijelovi programa trebaju ponoviti konačan broj puta. Osnovni oblik (sintaksa) for petlje:

**for (početno stanje; uslov; ažuriranje)**

```
{  
    blok naredbi;  
}
```

Svaka for petlja ima svoju kontrolnu varijablu. Kontrolnoj varijabli se na početku petlje pridružuje početna vrijednost. Vrijednost kontrolne varijable svakim se prolaskom kroz petlju mijenja za vrijednost ažuriranja (priraštaja). Petlja se izvršava sve dok je uslov ispunjen (logička istina; 1).

1. Potrebno je ispisati brojeve od 1 do 20. Ispis neka bude oblika:

Ispis brojeva od 1 do 20:

1 2 3 4 5 6 7 8 9 .....20

### Napomene:

Treba zadati početnu vrijednost kontrolne varijable petlje (brojac=1), postaviti uslov (brojac<=20) i zadati ažuriranje (brojac++).

Naredba koja se izvodi konačan broj puta je ispis trenutne vrijednosti varijable brojac.

2. Potrebno je dopuniti prethodni primjer tako da se ispisuju brojevi iz raspona od M do N (raspon bira korisnik). Ispis neka bude oblika:

Upisi pocetni broj raspona:

Upisi završni broj raspona:

Ispis brojeva od ... do ... :

... ..

3. Treba ispisati parne brojeve iz raspona od 50 do N. N bira korisnik. Ispis neka bude oblika:

Upisi završnu vrijednost raspona:

Parni brojevi iz raspona od 50 do ... su:

50 52 54 56 ...

4. Modifikovati prethodni program tako da ispisuje parne brojeve unazad; od 100 do N. N bira korisnik. Ispis neka bude oblika:

Upisi završnu vrijednost raspona:

Parni brojevi iz raspona od 100 do ... su:

100 98 96 94 ....

5. Treba ispisati brojeve iz raspona od M do N djeljive brojem B. Ispis neka bude oblika:

Upisi pocetnu vrijednost raspona:

Upisi završnu vrijednost raspona:

Upisi broj za provjeru djeljivosti:

Brojevi djeljivi s ... iz raspona od ... do ... su:

... ..

6. Napisati program koji izračunava sumu prvih N brojeva. (N unosi korisnik!).
7. Napisati program koji izračunava proizvod prvih N brojeva. (N unosi korisnik!).
8. Napisati program koji prikazuje sve brojeve djeljive sa 8 u intervalu od 300 do 400.
9. Napisati program koji izračunava i ispisuje sumu kvadrata prirodnih brojeva od k do n.
10. Napisati program koji izračunava i ispisuje sumu prirodnih brojeva u intervalu od 1 do n koji su djeljivi sa 7 i sa 3.
11. Napisati program koji izračunava sumu i proizvod svih brojeva djeljivih sa tri u intervalu od 1 do 300.
12. Napisati program koji izračunava proizvod neparnih brojeva od 1 do n.
13. Napisati program koji će sabrati sve cijele brojeve između 200 i 300 koji su dijeljivi sa 7. Koliko ima takvih brojeva?

14. Napisati program koji izračunava sumu prirodnih brojeva u intervalu od k do n čija je cifra jedinica 8. Tu sumu npr. čine brojevi 8,18,28....
15. Napisati program koji izračunava aritmetičku sredinu prvih n prirodnih brojeva koji su djeljivi sa 3.
16. Potrebno je ispisati sve trocifrene brojeve koji imaju cifru desetice djeljivu s 2, a cifru jedinice jednaku 7.
17. Potrebno je ispisati sve trocifrene brojeve čiji je zbir cifara 5, a zadnja im je cifra 0 (cifra jedinice).

#### **Napomena:**

Da bi se dobili svi trocifreni brojevi koje treba provjeriti, potrebno je napraviti petlju čija se vrijednost kontrolne varijable mijenja unutar potrebnog raspona:

početna vrijednost: (i=100), uslov: (i<=999), prirast: (i++).

Zatim iz svakog trocifrenog broja treba izdvojiti cifre jedinice, desetice i stotice. Kada se cifre određenog broja izdvoje, potrebno ih je sabrati ( $z=j+d+s$ ).

Svaki od brojeva koje treba provjeriti mora zadovoljiti dva uslova: zbir cifara mora biti jednak 5 ( $z==5$ ) a zadnja cifra mora biti jednaka 0 ( $j==0$ ). Stoga treba zadati složeni uslov upotrebom logičkog operatora I (AND).

Ako je složeni uslov `if((z==5)&&(j==0))` istinit, broj se ispisuje.