10. NAREDBE PONAVLJANJA (CIKLUSI)

Svi programi, bez obzira na složenost, predstavljaju neki tok izvršavanja naredbi. U većini programa se javljaju situacije kada je potrebno neku naredbu ili grupu naredbi izvršiti više puta. Ukoliko je naredbu potrebno izvršiti konačan broj puta, moguće je koristiti linijske struktura, sekvence, tako što bi se naredba jednostavno ponovila određeni broj puta. Međutim, ako je naredbu potrebno ponoviti veći broj puta, a taj broj može biti i promenljiv u zavisnosti od izvršavanja ostatka programa, u takvim slučajevima je neophodno uvesti ciklične strukture. **Ciklične strukure omogućavaju izvršavanje jedne ili više naredbi određeni broj puta**, pri čemu broj ponavljanja može biti definisan uslovom koji određuje kada se ponavljanje prekida. Ciklične strukture vrlo često nazivaju i iteracijama ili petljama. **Ciklusi su, znači, upravljačke strukture koje omogućavaju ponovljeno izvršavanje nekih naredbi.**

Po načinu zadavanja uslova koji treba da se ponavlja postoje:

- ciklusi koji traju **dok je uslov ispunjen**,
- ciklusi koji traju **dok uslov nije ispunjen**,
- **brojački ciklusi**.

U programskom jeziku C postoje tri upravljačke strukture za realizaciju ciklusa: **while**, **do-while** i **for**. Prve dve upravljačke strukture spadaju u cikluse koji traju dok je uslov ispunjen, dok for pripada brojačke cikluse.

WHILE ciklus - Ciklus sa preduslovom while omogućava organizovanje ciklusa sa nepoznatim brojem ponavljanja, koji zavisi od izračunavanja u telu ciklusa. Operator koji obrazuje telo ciklusa može biti prost ili složen. Osobine while ciklusa:

- While ciklus se izvršava sve dok je uslov logička istina (različit od nule),
- Koristi se kada se ne zna koliko će se ciklus izvršavati,
- Uvek se prvo proverava da li je uslov logička istina, te ako jeste naredba se izvršava,
- Može se desiti da se uslov ne izvrši nijednom (na početku uslov nije zadovoljen).

FOR ciklus – Najčešća primena je u formiranju brojačkih ciklusa, koji se koriste kada je poznato koliko puta će se ciklus izvršiti. Kod brojačkih ciklusa postoji uvek jedna kontrolna promenljiva čija se vrednost menja od neke početne pa do neke krajnje vrednosti. Na kraju svakog prolaska kroz ciklus vrednost kontrolne promenljive se promeni za neku stalnu vrednost, sve dok se ne stigne do krajnje vrednosti. For ciklus obuhvata sledeće radnje:

- Postavlja početne vrednosti promenljivih pre ulaska u ciklus,
- Proverava da li treba ulaziti u ciklus pre svakog prolaska kroz ciklus,
- Ako uslov na samom početku nije zadovoljen ciklus se neće izvršiti,
- Prilikom prolaska kroz ciklus vrši se menjanje jedne ili više promenljivih.

DO-WHILE ciklus - while i for ciklusi vrše proveru uslova na vrhu ciklusa, dok do-while petlja takvu proveru radi na kraju petlje, nakon prolaska kroz telo petlje, što znači da se mora izvršiti najmanje jednom. Osobine do-while ciklusa:

- Koristi se kada se ne zna koliko će se puta ciklus ponavljati,
- Prvo se izvršava naredba koja čini sadržaj ciklusa, a posle toga se izračunava,
- Vrednost logičkog izraza - uslova, ako se dobija logičko tačno, naredba se ponovno izvršava,
- Ciklus se završava kada uslov dobija vrednost logičke neistine,
- Ciklus se izvršava najmanje jednom.

11. WHILE CIKLUS

Ciklus sa preduslovom WHILE omogućava organizovanje ciklusa sa nepoznatim brojem ponavljanja, koji zavisi od izračunavanja u telu ciklusa. Operator koji obrazuje telo ciklusa može biti prost ili složen.

Osobine while ciklusa:

- While ciklus se izvršava sve dok je uslov logička istina (različit od nule),
- Koristi se kada se ne zna koliko će se ciklus izvršavati,
- Uvek se prvo proverava da li je uslov logička istina, te ako jeste naredba se izvršava,
- Može se desiti da se uslov ne izvrši nijednom (na početku uslov nije zadovoljen),
- Postoji mogućnost beskonačnog ponavljanja ciklusa, što rezultuje greškom u izvršavanju programa.

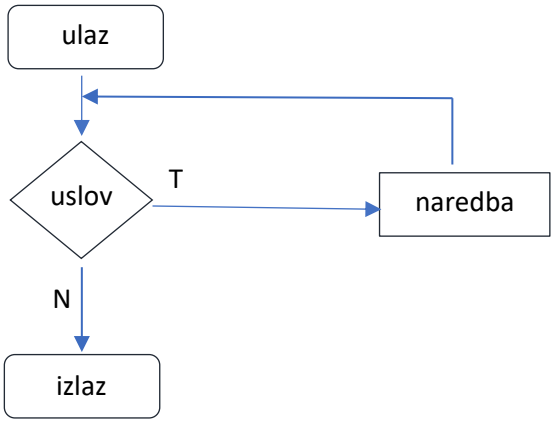
Opšti oblik while ciklusa je:

```
while (uslov) naredba;
```

ili sa blokom naredbi, tj. sekvencom:

```
while (uslov)
{
  naredba1;
  naredba2;
  …
  naredbaN;
}
```

Algoritamski prikaz **while** ciklusa:



Primer: Program koji izračunava sumu prvih n prirodnih brojeva.

```
#include <stdio.h>
int main()
{
   int i=1,s=0,n;
   printf("Unesite broj do kog se izracunava suma: ");
   scanf("%d", &n);
   while (i<=n)
      {
         s=s+i;
         i++;
      }
   printf("Suma iznosi: %d", s);
   return 0;
}
```

Rezultat izvršavanja:

```
Unesite broj do kog se izracunava suma: 143

Suma iznosi: 10296
```

12. FOR CIKLUS

Najčešća primena FOR ciklusa je u formiranju brojačkih ciklusa, koji se koriste kada je poznato koliko puta će se ciklus, tj. neki niz naredbi izvršiti. Kod brojačkih ciklusa postoji uvek kontrolna promenljiva čija se vrednost menja od neke početne pa do neke krajnje vrednosti. Na kraju svakog prolaska kroz ciklus, vrednost kontrolne promenljive se menja za neku stalnu vrednost, sve dok se ne stigne do krajnje vrednosti. Tada se ciklus završava i sledi izvršavanje prve sledeće naredbe nakon ciklusa.

For ciklus obuhvata sledeće **radnje**:

- Postavlja početne vrednosti promenljivih pre ulaska u ciklus.
- Proverava da li treba ulaziti u ciklus pre svakog prolaska kroz ciklus.
- Ako uslov na samom početku nije zadovoljen ciklus se neće izvršiti.
- Prilikom prolaska kroz ciklus vrši se menjanje vrednosti jedne ili više promenljivih ili složenijih struktura podataka.

Kod for ciklusa, za razliku od while i do-while, **ne postoji mogućnost beskonačnog ponavljanja** ciklusa.

Opšti oblik for ciklusa je:

```
for (izraz1;uslov;izraz2)
{
  naredba1;
  naredba2;
  …
  naredbaN;
};
```

U operatoru **for** se navode tri dela izraza koji upravljaju izvršavanjem ciklusa i razdvajaju se **;** .

izraz1 - Predstavlja pripremu za ulazak u ciklus, oblika je:
kontrolna_promenljiva = pocetna_vrednost

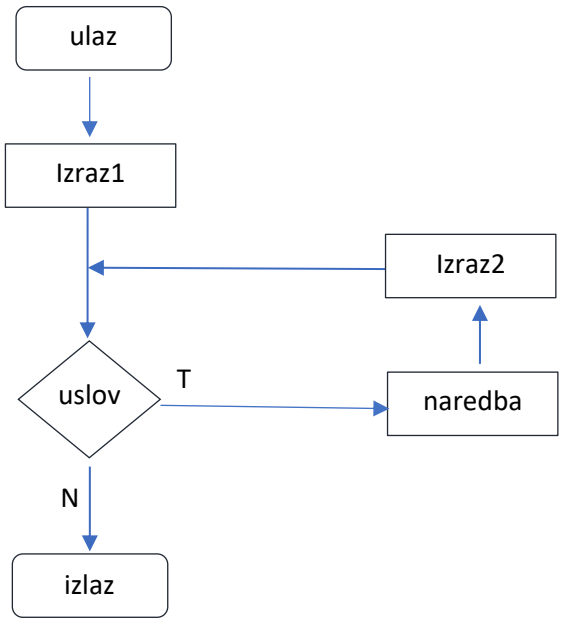
uslov - Logički izraz koji predstavlja uslov za nastavljanje ciklusa:
kontrolna_promenljiva relacijski operator(>,>=,<,<=) krajnja_vrednost

izraz2 - Obuhvata završne radnje na kraju svakog prolaska kroz ciklus ili pripremne radnje za prolazak kroz naredni ciklus:
kontrolna_promenljiva promena vrednosti (++, --, itd.)

Najčešće se kontrolna promenljiva na početku postavlja na jedan, a korak za koji se kontrolna promenljiva povećava jednaka je vrednosti jedan. Najčešći oblik for cilkusa kao brojačkog ciklusa je:
for (kontprom=1; kontprom<=krajnjavrednost; kontprom++) naredba;

Vrednost promenljive **krajnjavrednost** pokazuje nam koliko će se puta **for** ciklus izvršiti. Ako je vrednost promenljive **krajnjavrednost** manja od 1 **for** ciklus se neće nijednom izvršiti.

Algoritamski prikaz **FOR** ciklusa:



Primer: Program koji izračunava sumu prvih 100 prirodnih brojeva (Gausov zadatak).

```
#include <stdio.h>
int main()
{
   int i,s=0;
   for (i=0;i<=100;i++)
      {
         s=s+i;
      }
   printf("Suma prirodnih brojeva do 100 iznosi: %d", s);
   return 0;
}
```

Rezultat izvršavanja:

```
Suma prirodnih brojeva do 100 iznosi: 5050
```

13. DO-WHILE CIKLUS

Operator ciklusa sa postuslovom **DO-WHILE** je naredba ponavljanja koja proveru uslova za izvršavanje naredbi radi na kraju petlje, nakon prolaska kroz telo petlje, što znači da se mora izvršiti najmanje jednom. Ako je vrednost uslova ispunjena (tačna), telo ciklusa se ponavlja i ovaj postupak traje, dok uslov ne dobije vrednost netačno. Operator koji obrazuje telo ciklusa može biti prost ili složen (jedna naredba ili blok sastavljen od više naredbi).

Osobine do-while ciklusa:

- Koristi se kada se ne zna koliko će se puta ciklus ponavljati.
- Prvo se izvršava naredba koja čini sadržaj (telo) ciklusa, a posle toga se izračunava uslov.
- Vrednost logičkog izraza - uslova, ako se dobija logičko tačno, naredba se ponovno izvršava.
- Ciklus se završava kada uslov dobija vrednost logičke neistine.
- Ciklus se izvršava najmanje jednom.
- Postoji mogućnost beskonačnog ponavljanja ciklusa, što rezultuje greškom u izvršavanju programa.

Opšti oblik **do-while** naredbe je:

```
do{
  naredba;
} while (uslov);
```

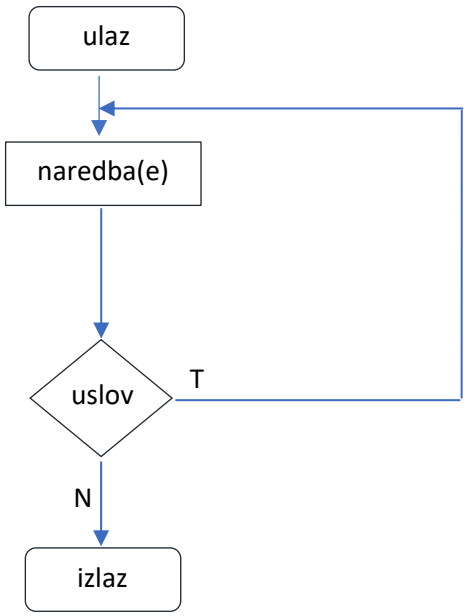
Ili sa blokom naredbi/sekvencom:

```
do {
  naredba1;
  naredba2;
  …
  naredbaN;
} while (uslov);
```

Naredba ili više naredbi u bloku se izvršava(ju), nakon čega se proverava uslov. U slučaju da je istinit, naredba(e) se ponovo izvršava(ju). Kada uslov postane neistinit, do-while petlja se završava.

Iskustveno je poznato da se ovaj ciklus koristi manje nego while i for, iako može biti izuzetno koristan u programiranju.

Algoritamski prikaz **DO-WHILE** ciklusa:



Primer: Ispis sume neparnih brojeva od 1 do n

```
#include <stdio.h>
int main ()
{
   int i=1,s=0,n;
   printf("Unesite prirodan broj: ");
   scanf("%d", &n);
   do
      {
         if(i%2==1){s=s+i;}
         i++;
      }
   while (i<=n);
   printf("Suma neparnih iznosi: %d", s);
   return 0;
}
```

Rezultat izvršavanja:

```
Unesite prirodan broj: 13
Suma neparnih iznosi: 49
```