

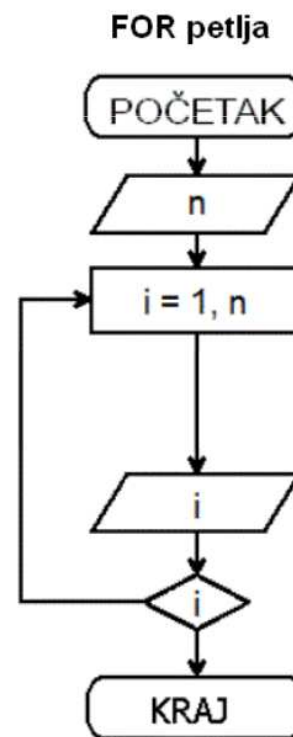
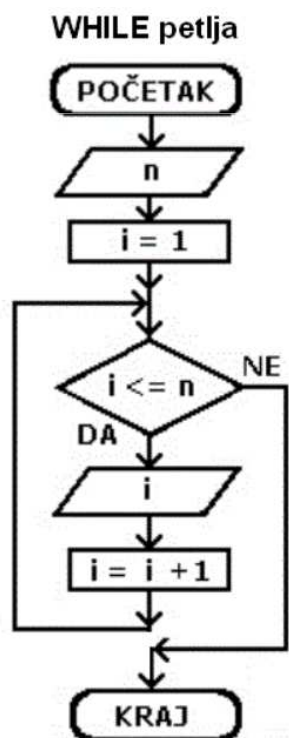
Ciklusi

predavač: Nadežda Jakšić

Programiranje programski jezik C

Ciklična struktura - petlja

u okviru **ciklične** strukture omogućeno je višestruko ponavljanje jedne ili više naredbi



While ciklus

sintaksa: **while (izraz)**
naredbe  naredbe koje čine telo ciklusa

- proveravanje uslova se vrši na vrhu ciklusa tj. izračunava se izraz
- ako izraz nema vrednost nula (tj. ako je uslov ispunjen), izvršava se jedna ili više naredbi koje zovemo telo ciklusa, a izraz se ponovo računa
- ciklus se nastavlja sve dok izraz ne dobije vrednost nula (uslov više nije ispunjen)
- tada se ciklus prekida i program nastavlja da se izvršava iza ciklusa

```
int i=1;
```

```
while (i<=5) //kada promenljiva i dostigne vrednost 6, izlazi se iz ciklusa  
{  
  printf ("Zdravo\n"); //ovaj ciklus 5 puta prikazuje reč Zdravo  
  i+=1; //koliko puta bi se izvršio ciklus kada bi i=0?  
} //šta bi se desilo kada ne bi bilo vitičastih zagrada?
```

For ciklus

sintaksa: **for (izraz1;izraz2;izraz3)**
naredbe

ovo je isto kao
izraz1;
while (izraz2)
{ naredbe
 izraz3;
}

osim za slučaj kada se pojavljuje i naredba **continue**

```
int i;  
for (i=1;i<=5;i++)  
printf ("Zdravo\n");
```

- najčešće, **izraz1** i **izraz3** su dodele ili pozivi funkcija, a **izraz2** je ispitivanje uslova
- nijedan od ovih izraza nije obavezan, ali se moraju pisati tačke-zarezi
- kao i kod **while** ciklusa, proveravanje uslova se vrši na vrhu ciklusa
- najčešće se koristi kada znamo tačno koliko puta će se izvršiti ciklus; zato se često zove **brojački ciklus**

Do while ciklus

- kod **while** i **for** ciklusa proveravanje uslova se vrši na vrhu, pa ako u prvom prolazu uslov nije ispunjen, izlazi se iz ciklusa i naredbe koje čine telo ciklusa se ne izvršavaju nijednom
- kod **do while** ciklusa, proverava se vrši na dnu ciklusa, nakon prolaska kroz telo ciklusa, što znači da će se naredbe koje čine telo ciklusa izvršiti makar jednom
- ovaj ciklus se koristi manje nego prethodna dva, ali ponekad ume da bude koristan

sintaksa:

do
naredbe
while (izraz);

```
int i=1; //reč Zdravo će biti prikazana pet puta, ali  
do      //ako bi promenljiva i na početku dobila  
{      //vrednost 56, reč bi ipak bila prikazana  
printf ("Zdravo\n"); //jednom  
i+=1;  
}  
while (i<=5);
```

Naredbe **break** i **continue**

- **break** omogućava programu trenutno napuštanje ciklusa i izlazak iz ciklusa (koristi se i u naredbi **switch** za prekid)
- **continue** je usko je povezana sa **break**, ali se ređe koristi; njena je uloga da otpočne sledeću iteraciju ciklusa u kome se program vrti, bez obzira o kojoj vrsti ciklusa se radi
- kod **while** i **do while** ciklusa primena naredbe **continue** znači da se tog trenutka prelazi na proveravanje uslova, a kod **for** ciklusa to znači da se prelazi nasledeći stepen inkrementacije (povećavanje vrednosti promenljive koja određuje koliko puta će se izvršiti ciklus)

Programi

//n puta se prikazuje reč Zdravo - **for** ciklus

```
int i,n;
printf ("Koliko puta želite da prikažete reč?\n");
scanf ("%d",&n);
for (i=1;i<=n;i++)
    printf ("Zdravo\n");
```

//n puta se prikazuje reč Zdravo - **while** ciklus

```
int i=1,n;
printf ("Koliko puta želite da prikažete reč?\n");
scanf ("%d",&n);
while (i<=n)
    { printf ("Zdravo\n");
      i+=1;}
```

Programi

//zbir parnih brojeva od 1 do 10 - **while** ciklus

```
int i=1; s=0;
while (i<=10)
{
    if (i%2==0) s+=i;
    i+=1;
}
printf ("\nZbir parnih brojeva u intervalu od jedan do deset je %d",s);
```

//zbir parnih brojeva od 1 do 10 – **for** ciklus

```
int i; s=0;
for (i=1;i<=10;i++)

    if (i%2==0) s+=i;

printf ("\nZbir parnih brojeva u intervalu od jedan do deset je %d",s);
```


Programi

//unosi se 5 celih brojeva i računa se zbir onih koji su parni

```
int i,broj,zbir=0;
for (i=1;i<=5;i++)
{
    printf ("Unesite broj\n");
    scanf ("%d",&broj);
    if (broj%2==0) zbir+=broj;
}
printf ("\nZbir parnih brojeva koje ste uneli je %d",zbir);
```

Programi

//aritmetička sredina 5 unetih brojeva

```
int i, broj,zbir=0;
float ars;
for (i=1;i<=5;i++)
{
    printf ("Unesite broj\n");
    scanf ("%d",&broj);
    zbir+=broj;
}
ars=(float)zbir/5;
printf ("\nAritmetička sredina unetih brojeva je %.2f",ars);
```

//ako je n unetih brojeva?

Programi

//ugnježdeni ciklusi (ciklus u ciklusu); usporavaju izvršavanje programa,
ali se ponekad ne mogu izbjeći
tablica množenja od 1 do 4

brojevi se ispisuju jedan ispod
drugog

```
int i, j;  
for (i=1;i<=4;i++)  
{  
    for (j=1;j<=4;j++)  
    {  
        printf ("%d * %d = %d\n",i,j,i*j);  
    }  
}
```

brojevi se ispisuju u tabeli 4x4

```
int i, j;  
for (i=1;i<=4;i++)  
{  
    for (j=1;j<=4;j++)  
    {  
        printf ("%d * %d = %d\t",i,j,i*j);  
    }  
    printf ("\n ");  
}
```

Programi

unos se 10 brojeva, pri čemu ne bi trebalo uneti broj 5; u slučaju da se unese broj 5 ispisuje se poruka **uneli ste broj 5, on se ne sabira i ne prebrojava**, i ide se na sledeći unos; na kraju se ispisuje broj ispravno unetih brojeva i njihov zbir

```
int i, br, zbir=0, broj=0;
for (i=1;i<=10;i++)
{
    printf ("Unesite broj\n");
    scanf ("%d",&br);
    if (br==5)
        { printf ("Uneli ste broj 5, on se ne sabira i ne prebrojava\n");
          continue;}
    broj+=1;
    zbir+=br;
}
printf ("Uneli ste %d ispravnih brojeva i njihov zbir je %d", broj, zbir);
```

Programi

unos se 10 brojeva, pri čemu ne bi trebalo uneti broj 5; u slučaju da se unese broj 5 ispisuje se poruka **prekid, uneli ste broj 5**; na kraju se ispisuje broj unetih brojeva do unosa broja 5 i njihov zbir

```
int i, br, zbir=0, broj=0;
for (i=1;i<=10;i++)
{
    printf ("Unesite broj\n");
    scanf ("%d",&br);
    if (br==5)
        { printf ("Prekid, uneli ste broj 5\n");
          break;}
    broj+=1;
    zbir+=br;
}
printf ("Uneli ste %d ispravnih brojeva\n", broj);
printf ("Njihov zbir je %d", zbir);
```

Programi

1. zbir parnih brojeva u intervalu od -50 do +100
2. koliko ima brojeva koji su deljivi sa tri u intervalu od **m** do **n** (interval unosi korisnik)
3. ispisati prirodne brojeve od 1 do 5 koji nisu deljivi sa 3
4. izračunati sumu prirodnih brojeva od **1** do **n** koji su deljivi sa **a**
5. program računa faktorijel unetog broja
6. ispisti sumu kvadrata parnih, a kubova neparnih brojeva u intervalu od **m** do **n**
7. da li je uneti broj palindrom (broj 123321 jeste palindrom)
8. korišćenjem ugnježenih ciklusa prikazati sve proste brojeve u opsegu od 2 do 50; broj je prost ako je veći od jedan i ako je deljiv samo jedinicom i samim sobom