

**Petlja for**

# Opšti oblik for naredbe

- **Naredba for koristi se u situacijam kada je unapred poznat broj ciklusa**
- **Naredba for ima sledeću strukturu:**

```
for(inicijalizacija; provera_uslova;korekcija)
naredba;
```

```
for(inicijalizacija; provera_uslova;korekcija)
{
naredba_1;
naredba_2;
.....
naredba_n;
}
```

# Objašnjenje mehanizma rada for ciklusa

- inicijalizacija predstavlja setovanje početne vrednosti brojača petlje (brojač petlje broji iteracije tj.cikluse)
- Ako `provera_uslova` daje rezultat “tačno” tj. predstavlja vrednost različitu od nule telo ciklusa se izvršava jedanput tj izvršava se naredba ili blok naredbi
- Zatim se izračunava vrednost korekcije , obično se vrši inkrementiranje brojača
- Zatim se ponovo vrši `provera_uslova`
- Ako `provera_uslova` ponovo da rezultat “tačno” ponovo se izvršava telo ciklusa ,....
- Iz ciklusa se izlazi kada `provera_uslova` da vrednost “netačno” tj. da vrednost jednaku 0

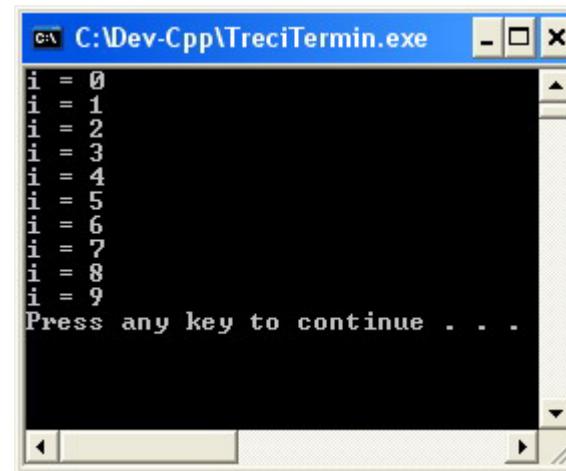
## Redosled izvršavanja zadatka u for ciklusu

- Inicijalizacije se izvršava samo jedanput
- Provera uslova za izvršavanje tela ciklusa se izvršava na početku svake iteracije
- Na kraju svake iteracije vrši se korekcija
- Pošto je for ciklus sa preduslovom moguće je da se telo ciklusa ne izvrši nijednom
- Telo ciklusa može biti samo jedna naredba ili blok naredbi (skup naredbi između litičastih zagrada)

# Primer upotrebe for petlje

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int i;
    for (i = 0; i < 10; i++)
        printf("i = %d\n",i);
    system("PAUSE");
    return 0;
}
```



```
C:\Dev-Cpp\TreciTermin.exe
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
Press any key to continue . . .
```

# Specijalni slučajevi ciklusa for

```
for(;provera_uslova;)
naredba;
```

ekvivalentno

```
while(provera_uslova)
naredba
```

```
int main(int argc, char *argv[])
{
    int i=0;
    for (;i < 10;)
    {
        printf("i = %d\n",i);
        i++;
    }
}
```

```
int i = 0, x;
for ( ; i != -1; )
{
    scanf( "%d", &i );
    .....
}
```

# Beskonačan ciklus

```
for(;;)  
naredba
```

ekvivalentan sa

```
while(1)  
naredba
```

```
int main(int argc, char *argv[])  
{  
    for(;;)  
    {  
        printf("Pozdrav!!!\n");  
    }  
}
```

# Naredba break

- **Koristi se za prevremeni završetak ciklusa**

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int i=0;
    for(;;)
    {
        if(i>9)
        break; // ako je brojac i veci od 9 izadji iz ciklusa
        printf("Pozdrav!!!\n");
        i++;
    }
}
```



# Primer upotrebe naredbe break

```
int main(int argc, char *argv[])
{
    int i;
    for(i=0;i<100;i++)
    {
        if(i>9)
            break; // ako je brojac i veci od 9 izađi iz ciklusa
        printf("i=%d\n",i);
    }
}
```

# Ugnježdavanje for naredbi

```
int main(int argc, char *argv[])  
{  
    int brVrsta,brKolona,i,j;  
    printf("ucitajte broj vrsta i kolona \n");  
    scanf("%d %d",&brVrsta,&brKolona);  
    printf("\n");
```

```
    for(i=0;i<brVrsta;i++)
```

```
    {
```

```
        for(j=0;j<brKolona;j++)
```

```
        {
```

```
            printf("*");
```

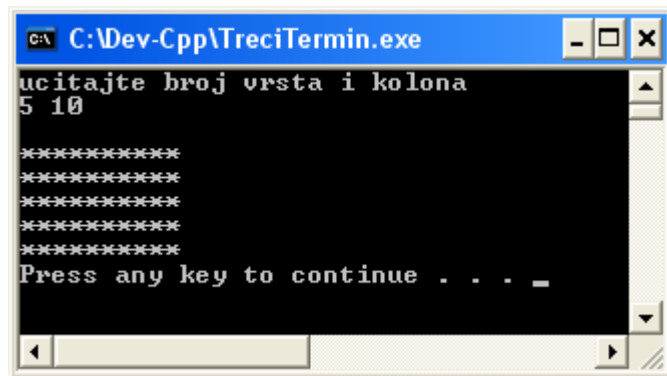
```
        }
```

```
    }  
    printf("\n");
```

```
    }
```

```
}
```

# Rezultat izvršavanje programa



```
C:\Dev-Cpp\TreciTermin.exe
ucitajte broj vrsta i kolona
5 10

*****
*****
*****
*****
*****
Press any key to continue . . . _
```