

## Ciklusi u programskom jeziku C (while, for)

### Ciklus while (ili while petlja)

Sintaksa while naredbe glasi:

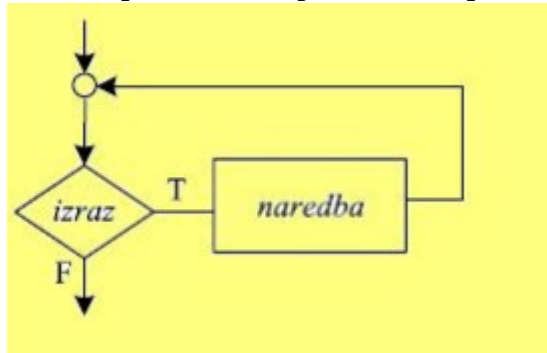
```
while ( izraz ) naredba
```

Na primer:

```
while (i<=10) { printf("%d ", i);}
```

Značenje je:

dok je (eng. while) izraz različit od nule, izvršava se naredba. Izraz predstavlja uslov ponavljanja while ciklusa.



Beskonačna petlja se može realizovati pomoću while petlje:

```
while(i=1) { printf("%d ", i);}
```

Program sa beskonačnom petljom

```
#include <stdio.h>
main()
{ int i;
  while(i=1) { printf("%d ", i);}
}
```

41.NCP koji ispisuje na standardni izlaz prvih 20 parnih brojeva. Upotrebiti while ciklus.

```
#include <stdio.h>
#define MAX 20
main ()
{
  int i; /*brojac trenutno ispisanih brojeva */
  i=1;
while ( i<=MAX)
  { printf("\\n%d", 2*i);
    ++i;
  }
}
```

42. NCP koji učitava pozitivan ceo broj sa standardnog ulaza i ispisuje najveću cifru tog broja.

ulaz izlaz

108 8

959 9

```
#include <stdio.h>
main()
```

```

{
short cifra=0, max=0; /*tekuca cifra broja, max cifra*/
unsigned a; /*ucitani broj*/
printf("Unesite broj: "); scanf("%u", &a);
    while(a!=0)
        {cifra=a%10;
          if (cifra >max) max=cifra;
          a/=10;
        }
printf("\nMax cifra je: %hd\n",max);
}

```

Zadaci za vezbanje (while)

1. Napisati program koji za uneto n (n>0) ispisuje prvih n prirodnih brojeva.  
Na primer: za ulaz 4 program ispisuje 1 2 3 4

2. Napisati program koji za uneto n (n>=0) na izlazu ispisuje faktorijel broja n.  
Na primer: za ulaz 4 izlaz je 24, za ulaz 5 izlaz je 120, za ulaz 0 izlaz je 1.

3. Napisati program koji za unete brojeve x i n (n>0) na izlazu ispisuje n-ti stepen broja x.  
Na primer: za ulaz 2 3 izlaz je 8, za ulaz 5 5 izlaz je 3125

4. Napisati program koji za uneti broj n (n>0) ispisuje zbir njegovih cifara.  
Na primer: za ulaz 123 izlaz je 6, za ulaz 5403 izlaz je 12

Resenje:

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

```

int main(){

    int n;          /* broj koji se unosi */
    int s=0;        /* zbir cifara */
    int c;          /* cifra koja se izdvaja */

    printf("Unesite broj n: ");
    scanf("%d", &n);

    if(n<0)
        printf("Uneli ste negativan broj!\n");
    else{

        while(n>0){
            c=n%10;
            s=s+c;
            n=n/10;
        }

        printf("Zbir cifara je: %d\n", s);
    }

    return 0;
}

```

5. Napisati program koji učitava sa standardnog ulaza prirodan broj manji od milijarde i na standardni izlaz ispisuje da li cifre tog broja obrazuju strogo rastući niz. Na primer, cifre broja 118 ne obrazuju strogo rastući niz, cifre broja 8876551 ne obrazuju strogo rastući niz, dok cifre broja 1234569 obrazuju strogo rastući niz.

<pre> PASCAL (za one koji su ucili pascal) Program P;   VAR a: longint;       b,c: integer; {tekuca cifra, prethodna cifra}       i: Boolean; { indikator rastuceg poretka } begin readln(a); i:=true; {na pocetku pretpostavimo da poredak je strogo rastuci} c:=a mod 10; { izdvajanje cifre jedinica broja a } a:=a div 10; while (a&lt;&gt;0) and i do begin b:=a MOD 10; a:=a DIV 10; i:=(b&lt;c); c:=b; end; if (i) then writeln(' Jeste strogo rastuci ') else writeln(' Nije strogo rastuci ');  end. </pre>	<pre> C #include &lt;stdio.h&gt; main () { long a; // broj sa stdin, int b,i,c; /* tekuca cifra, indikator rastuceg poretka, prethodna cifra*/ scanf("%ld",&amp;a); i=1; /* na pocetku pretpostavimo da poredak je strogo rastuci*/ c=a%10; /* izdvajanje cifre jedinica broja a */ a=a/10; while (a!=0 &amp;&amp; i!=0) { b=a%10; a=a/10; i=(b&lt;c); c=b; } (i==1)? printf("Jeste strogo rastuci\n") :printf("Nije strogo rastuci\n"); } </pre>
--	---

## Ciklus for (ili for petlja)



for-ciklus se zapisuje tako da se iza ključne reči for, zapišu u zagradama tri opciona izraza međusobno odvojena tačka-zarezom, a iza njih naredba koja čini telo petlje:

```
for ( izraz1opcion ; izraz2opcion ; izraz3opcion ) naredba
```

**Na primer:**

```
for(i=1;i<=n;i++){printf("%d",i);}
```

```
for(i=1,j=10;(i<=n) && (j>=0);i++, j--){printf("%d ",i); printf("%d ",2*j);}
```

```
for(i=1;i<=n;){printf("%d",i); i++;}
```

```
for(;i<=n;i++)printf("%d",i);
```

Beskonačna petlja se može realizovati pomoću for petlje:

```
for(;;)
{ /* beskonačna petlja */
}
```

Naredni for ciklus

```
for ( izrazlopcion ; izraz2opcion ; izraz3opcion ) naredba
se može zapisati kao sledeći while ciklus
izrazlopcion;
while (izraz2opcion){ naredba; izraz3opcion;}
```

Na primer:

<pre><b>for(i=1;i&lt;=n;i++){printf("%d",i);}</b></pre>	<pre>i=1; while(i&lt;=n) { <b>printf("%d",i);</b>   <b>i++;</b> }</pre>
<pre><b>for(i=1,j=10; (i&lt;=n) &amp;&amp; (j&gt;=0);i++, j--)</b> <b>{printf("%d ",i); printf("%d ",2*j);}</b></pre>	<pre>i=1; j=10; while(<b>(i&lt;=n) &amp;&amp; (j&gt;=0)</b>) { <b>printf("%d ",i);</b>   <b>printf("%d ",2*j);</b>   <b>i++; j--;</b> }</pre>

43. NCP koji ispisuje na standardni izlaz prvih 20 parnih brojeva. Upotrebiti for petlju (ciklus). Uporediti sa zadatkom 41.

Ideja: Prvih 20 parnih brojeva su elementi serije brojeva koja počinje brojem 2\*1, potom slede 2\*2, 2\*3, 2\*4,..., tj. to je serija od 20 brojeva u kojoj je svaki element 2 puta veći od članova serije {1, 2, 3, ..., 20}

```
#include <stdio.h>
#define MAX 20
main ()
{
int i; /*brojac trenutno ispisanih brojeva */
for (i=1; i<=MAX; ++i) printf("\n%d", 2*i);
}
```

44. NCP koji unosi 50 realnih brojeva sa standardnog ulaza, a ispisuje na standardni izlaz njihov zbir.

IDEJA: U ovom zadatku treba sabrati do 50 vrednosti, ali je nepraktično koristiti do 50 promenljivih za čuvanje unetih vrednosti, a zatim sabrati vrednosti tih 50 promenljivih. Zato se pribegava upotrebi ciklusa: tj. 50 puta će se vršiti učitavanje vrednosti sa tastature u promenljivu x, i nakon svakog učitavanja vrednost promenljive x će se dodati zbiru. Na početku je zbir jednak 0.

```
#include <stdio.h>
main ()
{
    int i=1; /*brojac ucitanih brojeva */
    float x, zbir; /* broj sa stdin, suma unetih brojeva */
    for(zbir=0, i=1; i<=50;i++) {scanf("%f",&x); zbir=zbir +x;}
    printf("\nZbir = %f\n", zbir);
}
```

45. NCP koji unosi nenegativan ceo broj sa standardnog ulaza, a ispisuje na standardni izlaz faktorijel tog broja. (  $0!=1$ ,  $n! = n * (n-1) * (n-2) * \dots * 2 * 1$  )

IDEJA: Racunanje se realizuje u n koraka. Promenljiva f uzima vrednosti sledecim redom

f=1 (za n=0 ili n=1)

f=2\*f=2\*1=2 (za n=2)

f=3\*f=3\*2=6 (za n=3)

f=4\*f=4\*6=24 (za n=4)

...

f=n\*f=n\* (n-1)!=n!

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

```
main ()
```

```
{
```

```
    int n; /*broj sa standardnog ulaza */
```

```
    long f=1; /*vrednost n!, uocite TIP long, 0!=1 */
```

```
    int i; /*brojac u ciklusu */
```

```
    printf("Unesite broj n: "); scanf("%d",&n);
```

```
    for(i=2; i<=n; i++) f=f*i;
```

```
    printf("\n%d! = %ld\n", n, f);
```

```
}
```

Zadaci za vezbanje (for)

1. Napisati program koji za uneto n ( $n>0$ ) ispisuje zbir prvih n prirodnih brojeva.

Na primer: za ulaz 4 program ispisuje 10

Rešenje:

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

```
int main(){
```

```
    int n,i;
```

```
    printf("Unesite broj n: ");
```

```
    scanf("%d", &n);
```

```
    if(n<0)
```

```
        printf("Uneli ste negativan broj!\n");
```

```
    else{
```

```
        printf("Prvih %d brojeva je:\n", n);
```

```
        for(i=1;i<=n;i++)
```

```
            printf("%d ", i);
```

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

```
    }
```

```
    return 0;
```

```
}
```

2. Napisati program koji za uneto n ( $n>0$ ) ispisuje zbir prvih n parnih prirodnih brojeva.

Na primer: za ulaz 4 program ispisuje 20

3. Napisati program koji za unete brojeve x i n ( $n>0$ ) na izlazu ispisuje n-ti stepen broja x.

Na primer: za ulaz 2 3 izlaz je 8, za ulaz 5 5 izlaz je 3125

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

```

int main(){
    int x, n;
    int i;
    int s;
    printf("Unesite brojeve x i n: ");
    scanf("%d %d", &x, &n);

    if(n<0)
        printf("Niste uneli dozvoljene vrednosti!\n");
    else{

        s=1;
        for(i=1;i<=n;i++)
            s=s*x;

        printf("Trazena vrednost je %d\n", s);
        // printf("Trazena vrednost je %f\n", pow(x,n));
        // poziv finkcije pow(x,n) zahteva ukljucivanje zaglavlja math.h

    }
    return 0;
}

```

4. Napisati program koji iscrtava jednakokraki pravougli trougao cija je stranica dimenzije n. Npr. za uneto n=4 iscrtava se

```

*
**
***
****

```

Resenje:

```

PASCAL (za one koji su ucili pascal)
Program P;
VAR n, i, j :integer;

begin
    writeln(' Unesite broj n: ');
    readln(n);

    for i:=1 to n do
    begin
        for j:=1 to i do
            write('*');
        writeln;
    end;

end.

```

```

C
#include <stdio.h>

int main(){

    int n, i, j;

    printf("Unesite broj n: ");
    scanf("%d", &n);

    for(i=1;i<=n;i++){
        for(j=1;j<=i;j++)
            printf("*");
        printf("\n");
    }

    return 0;
}

```

5. Napisati program koji za uneto n iscrtava kvadrat dimenzije n. Npr. za uneto n=6 iscrtava se

```

*****
*      *
*      *
*      *
*      *
*      *
*****

```

Ideja resenja:

```
/*
   ideja je da kvadrat koji treba iscrtati posmatramo kao jednu matricu sa
   n vrsta i n kolona; nase iscrtavanje ce se odvijati po vrstama:
   prva i poslednja vrsta sastoje se samo od n zvezdica,
   dok se vrste koje se nalaze izmedju, sastoje od jedne zvezde, n-2 belina i
   jos jedne zvezde;
   naravno, posle iscrtavanja svake vrste treba preci u naredni red;
*/

#include <stdio.h>

int main(void){

    int n;
    int k, v;
    printf("Unesite dimenziju kvadrata: ");
    scanf("%d", &n);

    /* dok vrsta v ide od 1 do n radimo sledece */
    for(v=1; v<=n; v++){

        /* ukoliko je u pitanju prva ili poslednja vrsta ispisujemo n zvezdica: po jednu u svakoj
        koloni */
        if(v==1 || v==n){
            for(k=1;k<=n;k++) printf("*");
            printf("\n");
        }

        else
        {
            /* ukoliko vrsta nije ni prva ni poslednja */
            /* iscrtavamo prvu zvezdu */
            printf("*");

            /* pa zatim iscrtavamo n-2 beline */
            for(k=1;k<=n-2; k++) printf(" ");

            /* i iscrtavamo poslednju zvezdu */
            printf("*");

            /* potrebno je postaviti jos i znak za novi red jer smo zavrшили sa iscrtavanjem vrste */
            printf("\n");
        }

    }

    return 0;
}
```

6. Napisati program koji za uneto  $n$  ( $n \geq 1$ ) iscrtava kvadrat kao na slici dimenzije  $n$ . Npr. za uneto  $n=6$  iscrtava se

```
*****
**      *
*  *    *
*   *  *
*    **
*****
```

7. Napisati program koji za uneto  $n$  ( $n \geq 1$ ) iscrtava slovo X visine  $2n+1$ . Npr. za uneto  $n=2$  iscrtava se

```
  *   *
 * *
  *
 * *
 *   *
```

Resenje:

```
#include <stdio.h>

int main(){

    int n, i, j;

    printf("Unesite broj n: ");
    scanf("%d", &n);

    if(n<1){
        printf("Niste uneli korektnu vrednost!\n");
        return 0;
    }

    for(i=1;i<=2*n+1;i++){
        for(j=1;j<=2*n+1;j++){
            if(i==j || i+j==2*n+2)
                putchar('*');
            else
                putchar(' ');
        }
        printf("\n");
    }

    return 0;
}
```

8. Napisati C program koji učitava sa standardnog ulaza prirodan broj manji od 100000 i na standardni izlaz ispisi da li je taj broj Armstrongov. N-to cifren broj je Armstrongov ako je jednak sumu  $N$ -tih stepena svojih cifara. Na primer, 1002 nije, 370 jeste, 407 jeste.

```
#include <stdio.h>
main()
{
    long x; /* broj sa stdin */
    short n=0; /* broj cifara broja x */
    long stepen; /* n-ti stepen cifre broja x */
    long k1=0,k; /* suma n-tih stepena cifara broja x, pomocna kopija broja
x */

    int j; /* brojacka promenljiva */
    printf("Unesite broj:");
    scanf("%ld",&x);
    /* prebrojavanje broja cifara broja x */
    k=x;
    while(k!=0)
    {
        n++;
        k/=10;
    }
```



```

/* racunanje i-tih stepena broja x*/
k=x;
while (k!=0)
{
    for (stepen=1,j=1; j<=n;j++) stepen*=(k%10); //stepen=pow(k%10,n);
    k1+=stepen;
    k/=10;
}
if (k1==x) printf("\nJeste Armstrongov broj");
else printf("\nNije Armstrongov broj");
}

```

9. Napisati C program koji učitava sa standardnog ulaza prirodan broj  $n$  i na standardni izlaz ispisuje prvih  $n$  Nivenovih brojeva. Prirodan broj je Nivenov ako je deljiv sumom svojih cifara. Na primer, za  $n=15$ , izlaz je 1,2,3,4,5,6,7,8,9,10,12,18,20,21,24.

Resenje:

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

```
main()
```

```
{ int n,s,k,p,m; /*dimenzija ispisa, suma cifara, kopija broja kandidat,
kandidat za Nivenov broj, broj ispisanih brojeva*/
```

```
scanf("%d",&n);
```

```
for (k=1,s=0,m=1;m<=n;)
```

```
{ p=k;
```

```
/*racunanje sume cifara broja k*/
```

```
while (k!=0)
```

```
{ s+=k%10;
```

```
k/=10;
```

```
}
```

```
if (p%s==0) { printf("%d ",p);m+=1; } /*ispis Nivenovog broja */
```

```
k=p+1; s=0;
```

```
}
```

```
}
```

10. Napisati C program koji učitava sa standardnog ulaza datum u formatu dd.mm.gggg i ispisuje na standardni izlaz redni broj tog dana u godini. Pretpostaviti da datumi su korektno uneti i da se odnose na vreme posle 15.10.1582.

Na primer	ulaz	01.01.2012.	izlaz	1
	ulaz	03.02.2011.	izlaz	34
	ulaz	01.03.2012.	izlaz	61
	ulaz	01.03.2011.	izlaz	60

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

```
main() {
```

```
int d,m,g; /*ucitani dan, mesec, godina*/
```

```
short gp; /*indikator da li je godina prestupna*/
```

```
int k,m1; /*redni broj dana, redni broj meseca koji prethode mesecu m*/
```

```
int br; /* broj dana meseca m1 godine g*/
```

```
scanf("%d.%d.%d",&d,&m,&g);
```

```
m1=1;k=0;
```

```
while (m>m1){
```

```
switch(m1){
```

```
case 4:case 6:case 9:case 11:br=30;break;
```

```
case 2:gp=((g%4==0) && (g%100!=0)) || (g%400==0); br=28+gp;break;
```

```
        default:br=31;
    }
    k+=br;
    m1++;
}
k+=d;
printf("%d",k);
}
```