

TEORETSKI DEO-III blok

NCP=Napisati C program

Naredba switch

```
switch(iskaz) {  
case konst1: iskaz1; break;  
case konst2: iskaz2; break;  
case konst100: iskaz100; break;  
default: iskaz101;  
}
```

```
if (iskaz==konst1) iskaz1;  
else if (iskaz==konst2) iskaz2;  
  
else if (iskaz==konst100) iskaz100;  
else iskaz 101;
```

Primer1:

```
switch(mesto) {  
case 1: printf("\nZlato"); break;  
case 2: printf("\nSrebro"); break;  
case 3: printf("\nBronza"); break;  
default: printf("\nBez medalje");  
}
```

```
if (mesto==1) printf("\nZlato");  
else if (mesto==2) printf("\nSrebro");  
else if (mesto==3) printf("\nBronza");  
else printf("\nBez medalje");
```

```
switch(iskaz) {  
case konst1:  
case konst2:  
case konst100: iskaz100; break;  
default: iskaz101;  
}
```

```
if ((iskaz==konst1) ||(iskaz==konst2) ||  
(iskaz==konst100))  
iskaz100;  
else iskaz101;
```

Primer2:

```
switch(mesto) {  
case 1: case 2: case 3:  
printf("\nMedalja"); break;  
default: printf("\nBez medalje");  
}
```

```
if ((mesto==1) ||(mesto==2) ||(mesto==3))  
printf("\nMedalja");  
else printf("\nBez medalje");
```

39.NCP koji učitava dva cela broja i koristeći switch naredbu ispisuje njihov maksimum.

```
#include <stdio.h>  
main()  
{  
    int a,b, max;  
    scanf("%d%d", &a, &b);  
    switch(a>b)  
    {  
        case 0:max=b;break;  
        case 1:max=a;  
    }  
    printf("\nmax=%d\n", max);  
}
```

40.NCP koji učitava malo slovo, a na standardni izlaz ispisuje da li je učitano slovo samoglasnik a,e,i,o,u.

```
#include <stdio.h>  
main()  
{  
    char x;  
    scanf("%c", &x);  
    switch(x)  
    {  
        case 'a': case 'e': case 'i': case 'o': case 'u':  
            printf("%c je samoglasnik\n", x); break;  
        case 'r': printf("slovo r\n"); break;  
        default: printf("%c je suglasnik\n", x);  
    }  
}
```

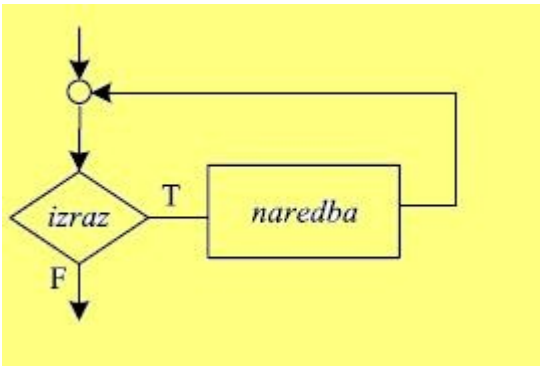
}

Ciklus while (ili while petlja)

Sintaksa while naredbe glasi:

while (izraz) naredba

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



41. NCP koji ispisuje na standardni izlaz prvih 20 parnih brojeva. Upotrebite 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;
    }
    printf("\nKRAJ\n");
}
```

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);
}

```

Četvrti domaći zadatak (zadaci 8,9)

8. (3 poena) NCP koji učitava sa standardnog ulaza prirodan broj i na standardni izlaz ispisuje sumu cifara tog broja.

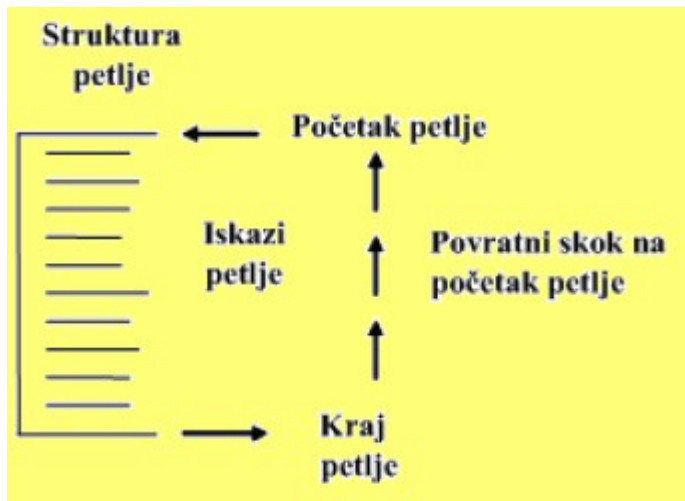
| | |
|------|-----------------|
| ULAZ | IZLAZ |
| 8 | 8 |
| 1028 | 11 |
| -55 | nekorektan ulaz |

9. (3 poena) NCP koji učitava sa standardnog ulaza prirodan broj i na standardni izlaz ispisuje netrivialne faktore (delioce) tog broja. Trivijalni faktori broja n su 1, n .

| | |
|------|-------|
| ULAZ | IZLAZ |
|------|-------|

| | |
|----|---------------|
| 1 | / |
| 7 | / |
| 21 | 3 7 |
| 30 | 2 3 5 6 10 15 |

Ciklus for (ili for petlja)



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

for (izraz1opcion ; izraz2opcion ; izraz3opcion) naredba

Beskonačna petlja se može realizovati pomoću for petlje:

for(;;)

```
{
/* beskonačna petlja */
}
```

Naredni for ciklus

for (izraz1opcion ; izraz2opcion ; izraz3opcion) naredba

se može zapisati kao sledeći while ciklus

izraz1opcion;

while (izraz2opcion){ naredba; izraz3opcion;}

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

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 $\{1, 2, 3, \dots, 20\}$

```
#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);
}

```

Ciklus do-while (ili do-while petlja)

Sintakse do-while naredbe je **do** naredba **while** (izraz) ;

Izraz predstavlja uslov za ponavljanje ciklusa.

Značenje je: izvrši naredbu, a zatim ponavlja tu naredbu dok je izraz logički istinit.

Naredbe u telu se petlje izvršavaju **bar jedan put**.

46. NCP koji ispisuje na standardni izlaz prvih 20 parnih brojeva. Upotrebiti do-while ciklus.

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

```

#define MAX 20

main ()
{
    int i=1; /*brojac trenutno ispisanih brojeva */

    do
    {
        printf("\n%d", 2*i);
        ++i;
    }while ( i<=MAX);

}

```

47. NCP koji unosi broj sa standardnog ulaza, formira broj sa ciframa u obrnutom poretku i ispisuje ga na standardni izlaz (npr. Ako se unese broj 248, program ispisuje 842)

```

#include <stdio.h>
main()

```

```
{
int n,t=0; /* n=broj sa stdin, t=vrednost broj sa
obrnutim ciframa */

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

/*formiranje vrednosti broja sa obrnutim ciframa*/
do
{
t=t*10+n%10; /* Hornerova shema */
n/=10;
}while(n);

printf("Novi broj je %d\n", t);

}
```