

STAVKI PONAVLJANJA

Krmilni stavki v C-ju so:
DO-WHILE stavek,
WHILE stavek in
FOR stavek.

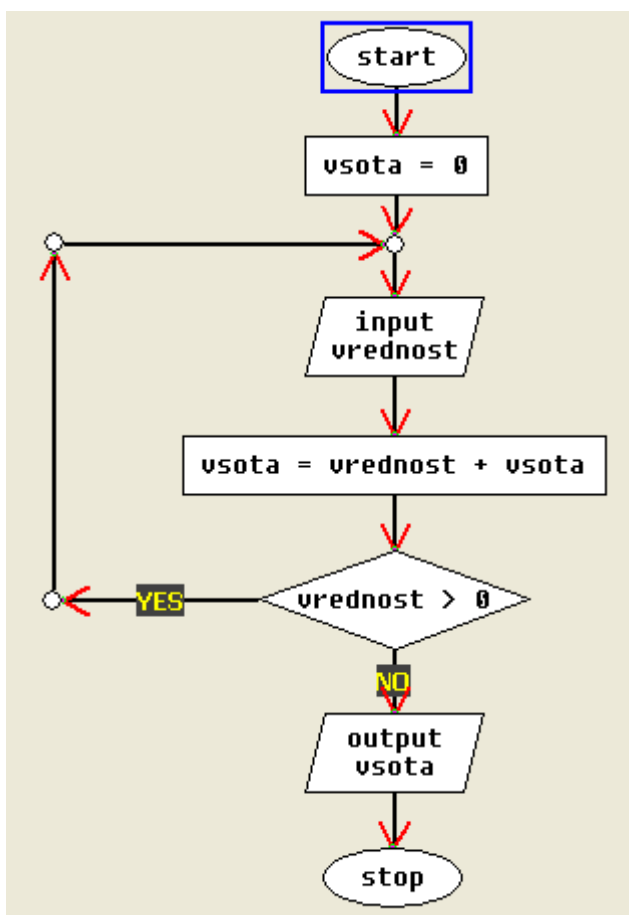
DO-WHILE STAVEK

```
do  
    stavki;  
while (pogoj);
```

STAVEK DO-WHILE UPORABIMO ZA PONAVLJANJE STAVKOV, KO SE LE-TI VSAJ ENKRAT ALI VEČKRAT IZVRŠIJO IN NE VEMO KOLIKOKRAT.

PRIMER: V življenju moramo določeno akcijo večkrat ponoviti. Na blagajni seštevamo vrednosti izdanih računov. Ko vtipkamo za vrednost računa 0, končamo s seštevanjem in izpišemo vsoto. Vsoto izračunavamo z delnimi vsotami. Algoritem je enak in ga izvajamo ljudje, ko seštevamo števila s prištevanjem.

Grafično prikazano:



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

int main(void)
{
    int vrednost, vsota;
    vsota = 0;
    do
    {
        printf("Vrednost: ");
        scanf("%d",&vrednost);
        vsota = vrednost + vsota;
    } while (vrednost != 0);
    printf("Vsota je %d\n",vsota);

    return 0;
}
```

Staveki se izvedejo najmanj enkrat. Iz zanke pademo takrat, ko je pogoj neizpolnjen.

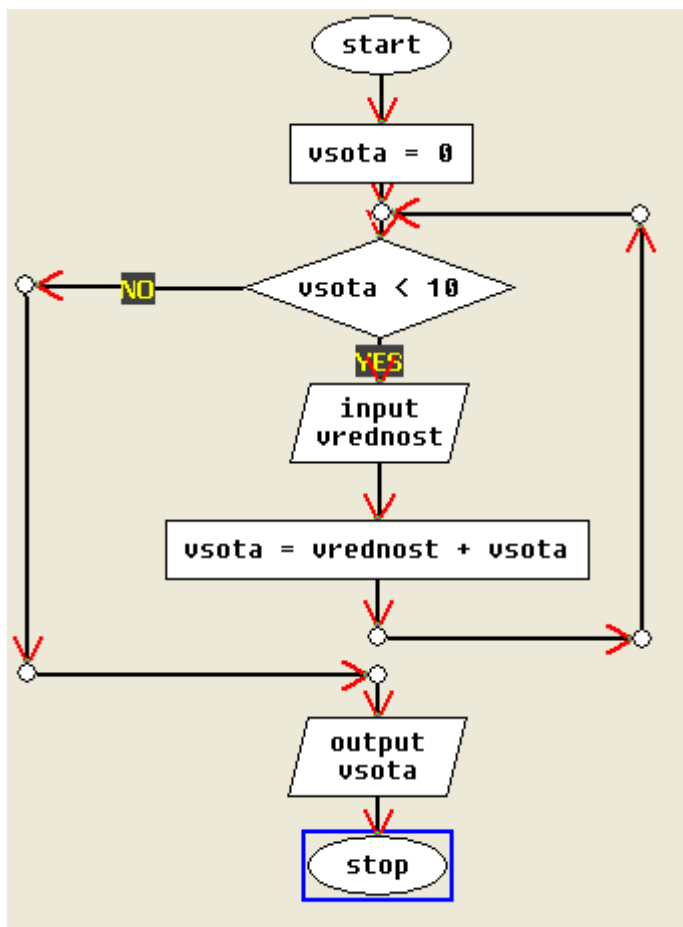
WHILE-DO ZANKA

while (pogoj)
stavki;

STAVEK WHILE UPORABIMO ZA PONAVLJANJE STAVKOV, KO SE LAHKO LE-TI NIČKRAT, ENKRAT ALI VEČKRAT PONOVIJO IN NE VEMO, KOLIKOKRAT.

PRIMER: Janez rad prosjači. V ta namen, želi napisati program, ki bo prebral, koliko tolarjev potrebuje in koliko tolarjev ima. Nato naj računalnik toliko dolgo povprašuje po dodatnih prispevkih, ki jih bo Janez naprosjačil, da si bo lahko kupil želeno stvar.

Grafično prikazano:



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

int main(void)
{
    int vrednost, vsota;
    vsota = 0;
    while (vsota < 10)
    {
        printf("Vrednost: ");
        scanf("%d", &vrednost);
        vsota = vrednost + vsota;
    }
    printf("Vsota je %d\n", vsota);

    return 0;
}
```

WHILE zanko lahko vedno nadomestimo z DO-WHILE ZANKO in obratno.

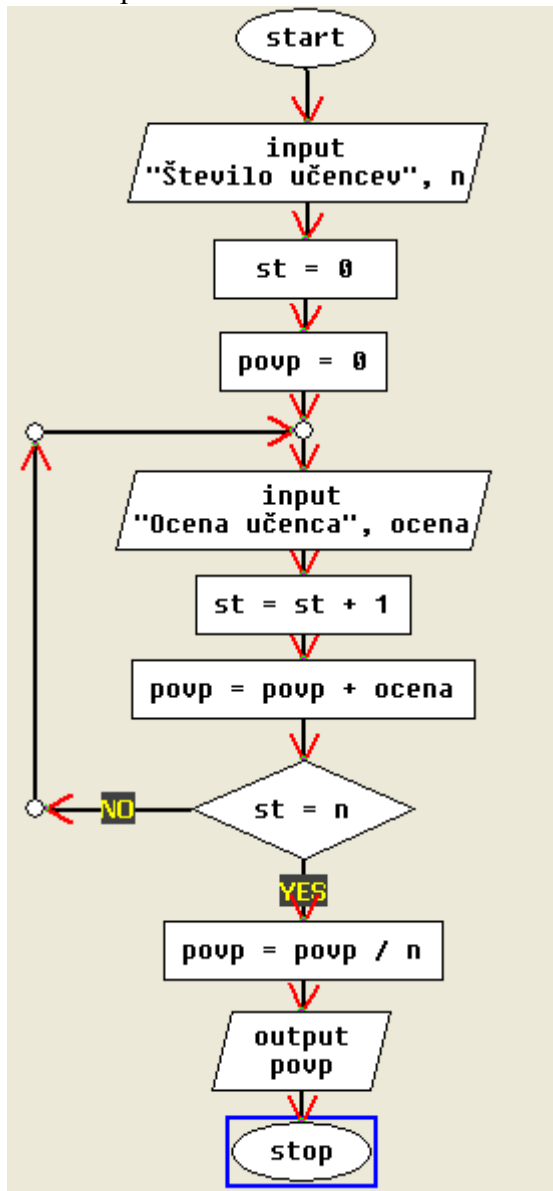
FOR ZANKA

for (inicializacija spremenljivk-števcev; pogoji ponavljanja; povečevanje števcov)
stavki;

STAVEK PONAVLJANJA FOR UPORABIMO, KO NATANČNO VEMO, KOLIKOKRAT SE MORAJO STAVKI PONOVI.

PRIMER: Izračunajmo povprečno oceno računalniškega testa. Uporabili bomo stavek ponavljanja for za seštevanje in izračunavanje povprečne ocene.

Grafično prikazano:



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

int main(void)
{
    int n, st, povp, ocena;
    povp = 0;
    printf("Število uencev: ");
    scanf("%d", &n);
    for (st = 0; st < n; st++)
    {
        printf("Ocena učenca: ");
        scanf("%d", &ocena);
        povp = povp + ocena;
    }
    povp = povp / n;
    printf("Povprečna ocena je %d\n", povp);

    return 0;
}
```

V večini primerov lahko zanko zapišemo ali z DO-WHILE ali z WHILE ali z FOR zanko.

Neskončna zanka:

```
for ( ; ; )
{
}
}
```

Continue in break ukaza

Continue in break nam predčasno ustavita izvajanje zanke. Ukaza delujeta enako v vseh treh zankah, ki jih programski jezik C pozna. Poglejmo si razliko na primeru.

Primer:

```
for (i = 1; i <= 10; i++)
{
    if (i == 5)
        continue;
    printf("%d ", i);
}
```

v tem primeru nam program izpiše: 1 2 3 4 6 7 8 9 10

Ko uporabimo stavek continue nam program preskoči izvajanje zanke in nadaljuje z izvajanjem naslednje iteracije zanke.

```
for (i = 1; i <= 10; i++)
{
    if (i == 5)
        break;
    printf("%d ", i);
}
```

v drugem primeru nam program izpiše: 1 2 3 4

Z uporabo break stavka nam program permanentno prekine izvajanje zanke. Program se nadaljuje z stavki za zanko.

NALOGE

1. Napišite program, ki prebere dve vrednosti. Prvo vrednost naj sešteje tolikokrat, kolikor smo vnesli drugo vrednost.
2. Napiši program, ki bo prebral 30 števil in seštel vsa pozitivna in vsa negativna števila, ter izpisal obe vsoti. Izračuna in izpiše naj tudi povprečno vrednost vseh števil.
3. Napiši program, ki bo izpisal prvi člen geometrijskega zaporedja, ki je večji od 1000. Geometrijsko zaporedje je: 1,2,4,8,16,...
4. Napišite program, ki bo bral števila, dokler ne vpišemo števila 0, ter poiskal in izpisal najmanjšega.
5. Izdelaj program, ki iz prebranega števila preko tipkovnice izpiše število števk.
6. Izpišite vsa praštevila od 1 do 100!
7. Izdelaj program, ki z uporabo for zanke izdela križec na sredini zaslona
8. Izdelaj program, ki z uporabo for zanke izdela kvadrat na sredini zaslona
9. Izdelaj program, ki z uporabo for zanke izdela krog na sredini zaslona
10. Izdelaj programe, ki z uporabo do-while, while do zanke izdela križec, kvadrat in krog na sredini zaslona.