

# Pogojni stavki

Pogosto želimo, da računalnik izvaja drugačno kodo glede na vrednost ene ali večih spremenljivk, npr. da nam pokaže drugačno vsebino, če smo napisali pravilno ali napačno geslo, da računalno sešteva, če smo pritisnili gumb za seštevanje, oz. odšteva, če smo pritisnili gumb za odštevanje, ipd. Z drugimi besedami, želimo upravljati potek programa (torej izbrati, katera koda naj se izvede) glede na vrednosti spremenljivk. Angleško takemu upravljanju pravimo *control flow*, najpogosteje pa ga izvajamo s t.i. *pogojnimi* ali *if stavki*. Osnovna struktura je sledeča:

## Primer

```
if (pogoj) {  
    // koda, ki se izvede, ce pogoj velja  
} else {  
    // koda, ki se izvede, ce pogoj ne velja  
}
```

*Pogoj* je nov pojem. Označuje neke vrste račun, katerega rezultat ni število, vendar *logična vrednost*. Tu sta možni vrednosti le dve: pravilno (angl. `true`) in napačno (angl. `false`). Če bo rezultat računa, navedenega v običajnih oklepajih v zgornjem `if` stavku, `true`, se bo izvedla koda znotraj prvih zavitih oklepajev, če pa je rezultat računa `false`, pa se bo izvedla koda v drugih zavitih oklepajih (tistih za besedo `else`). Drugega dela, t.j. `else` in oklepaji za njim, ni treba pisati, če ga ne želimo.

Kako pa zapišemo pogoj? Za to uporabimo posebne logične operatorje. Pri delu s številkami so nam na voljo naslednji:

- `==`: primerja dve številski vrednosti. Rezultat je `true`, če sta vrednosti enaki.
- `!=`: primerja dve številski vrednosti. Rezultat je `true`, če sta vrednosti različni.
- `<`: primerja dve številski vrednosti. Rezultat je `true`, če je vrednost na levi manjša od vrednosti na desni.
- `>`: deluje podobno kot `<`, le da v drugo smer; rezultat je `true`, če je vrednost na desni manjša od vrednosti na levi.
- `<=`: primerja dve številski vrednosti. Rezultat je `true`, če sta vrednosti enaki, ali če je vrednost na levi manjša od vrednosti na desni.
- `>=`: deluje podobno kot `<=`, le da v drugo smer.

## Primer

Poglejmo si primer uporabe pogojnega stavka.

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d%d", &a, &b);

    if (a == b) {
        printf("Stevili sta enaki.\n");
    }
    if (a != b) {
        printf("Stevili sta razlicni.\n");
    }
    if (a < b) {
        printf("Prvo stevilo je manjse od drugega.\n");
    }
    if (a > b) {
        printf("Prvo stevilo je vecje od drugega.\n");
    }
    if (a <= b) {
        printf("Prvo stevilo je manjse ali enako drugemu.\n");
    }
    if (a >= b) {
        printf("Prvo stevilo je vecje ali enako drugemu.\n");
    }
    return 0;
}
```

### Primer vhoda in izhoda

12 12

Stevili sta enaki.  
Prvo stevilo je manjse ali enako drugemu.  
Prvo stevilo je vecje ali enako drugemu.

### Primer vhoda in izhoda

3 7

Stevili sta razlicni.  
Prvo stevilo je manjse od drugega.  
Prvo stevilo je manjse ali enako drugemu.

### Pogoste napake

Pri primerjavi moramo uporabiti dva enačaja (==). Če uporabimo samo en enačaj, kot v matematiki (torej =), se bo program sicer zagnal, vendar ne bo deloval pravilno. Enojnega enačaja nikoli ne uporabljamo v pogoju `if` stavka.

### Pogoste napake

Taka primerjava deluje samo na številih. Če želimo primerjati dve besedili, za to potrebujemo posebno funkcijo `strcmp`, ki jo bomo podrobneje spoznali, ko bomo govorili o besedilu.

### Primer

Primer uporabe stavka `else`. Program bo uporabnika vprašal za PIN, in mu napisal, če je bil PIN pravilen oziroma napačen.

```
#include <stdio.h>

int main() {
    // Vprasaj uporabnika za PIN, in preveri, ce je pravilen
    int pin;
    scanf("%d", &pin);
    if (pin == 42) {
        printf("PIN je pravilen!");
    } else {
        printf("PIN je napacen!");
    }
    return 0;
}
```

#### Primer vhoda in izhoda

42

PIN je pravilen!

#### Primer vhoda in izhoda

7

PIN je napacen!

## Primer

if stavke lahko tudi gnezdimo; torej vstavimo enega v drugega. Spodnji program od uporabnika sprejme naročilo v restavraciji, kjer ponujajo dve vrsti hrane; juhe in sendviče. Na voljo sta dve vrsti juhe, in dve vrsti sendvičev.

```
#include <stdio.h>

int main() {
    // Uporabnik naj napise 1, ce zeli juho, in 2, ce zeli sendvic.
    int zelja;
    scanf("%d", &zelja);

    if (zelja == 1) {
        // Uporabnik naj napise 1, ce zeli goveja juho,
        // in 2, ce zeli paradiznikovo.
        scanf("%d", &zelja);
        if (zelja == 1) {
            printf("Ena goveja juha. Dober tek!\n");
        } else {
            printf("Ena paradiznikova juha. Dober tek!\n");
        }
    } else {
        // Uporabnik naj napise 1, ce zeli sendvic s sunko,
        // in 2, ce zeli vegeterjanski sendvic.
        scanf("%d", &zelja);
        if (zelja == 1) {
            printf("En sendvic s sunko. Dober tek!\n");
        } else {
            printf("En vegeterjanski sendvic. Dober tek!\n");
        }
    }
    return 0;
}
```

### Primer vhoda in izhoda

```
2
1
```

---

```
En sendvic s sunko. Dober tek!
```

## Pogoste napake

Pozorni bodimo tudi na postavitev kode. Običajno kodo znotraj zavutih oklepajev `if` stavka pišemo tako, da je poravnana štiri presledke bolj desno od kode zunaj `if` stavka. V nekaterih programskih jezikih je taka poravnava obvezna, v C/C++ pa ne, vendar nezamaknjena koda že v zelo majhnih programih postane popolnoma nepregledna. Branje in popravljanje kode je veliko lažje, če del kode znotraj zavutih oklepajev zamaknemo za štiri presledke. Za to lahko uporabimo tudi tipko `Tab`, ki se na tipkovnici nahaja levo od tipke `Q`.

Tako **nikoli** ne pišemo:

```
#include <stdio.h>
int main() {
if (3 > 2) {
printf("Velja\n");
} else {
printf("Ne velja\n");
}
return 0;
}
```