

Pogojni stavek

Pogojni stavek uporabimo, kadar želimo izvesti določen stavek samo v primeru, ko je izpolnjen nek pogoj. Pogojni stavek zapišemo z besedo `if`, ki ji sledi pogoj, temu pa stavek, ki ga želimo izvesti, če je pogoj izpolnjen (če ima od 0 različno vrednost).

```
if (pogoj) stavek;
```

Poglejmo si primer, kako lahko izračunamo absolutno vrednost danega števila. Recimo, da je število zapisano v spremenljivki `x`. Če je negativno, mu moramo spremeniti predznak.

```
if (x < 0) x = -x;
```

Poglejmo si primer izpisa ure v standardni obliki, npr: 9:15 ali 23:07 (pomembna je morebitna vodilna ničla pri minutah). Recimo, da imamo uro in minute shranjeni v celoštevilskih spremenljivkah `ura` in `min`. Najprej bomo izpisali uro in dvopičje. Če so minute enomestno število (če je minut manj kot 10), moramo izpisati še ničlo. Na koncu izpiše minute in to je vse. Opozoriti je treba, da se da to izpisati tudi bistveno bolj elegantno (brez uporabe pogojnega stavka), če bi poznali formatna določila v podrobnosti (več o tem kasneje).

```
printf("%d:", ura);  
if (min < 10) printf("0");  
printf("%d", min);
```

Če želimo v primeru, ko pogoj ni izpolnjen (ko ima vrednost 0), izvesti kakšen drug stavek, na koncu (običajno v novi vrsti) napišemo še besedo `else` ter stavek, ki ga želimo izvesti.

```
if (pogoj) stavek1;  
else stavek2;
```

Stavek, ki ga želimo izvesti, če je nek pogoj izpolnjen, je lahko sestavljen tudi iz več stavkov. V tem primeru stavke zapišemo enega za drugim in jih obdamo z zavirami oklepaji, kot je to prikazano na primeru. Takemu stavku rečemo tudi *sestavljeni stavek* ali *blok*. Stavke znotraj bloka pišemo z večjim odmikom od levega roba.

```
if (pogoj) {  
    stavek1;  
    stavek2;  
    stavek3;  
}  
else {  
    stavek4;  
    stavek5;  
}
```

Tudi pogojni stavek je čisto navaden stavek, torej ga lahko uporabimo kot enega od stavkov znotraj drugega pogojnega stavka (*gnezdenje* pogojnih stavkov). Pri tem pa lahko nastopijo nekatere težave. Če napišemo dva pogojna stavka (enega v drugem) in samo en `else`, včasih nastopijo dvomi, kateremu pogojnemu stavku ta `else` pripada. Pravila prevajalnika so sicer jasna (`else` pripada zadnjemu pogojnemu stavku brez njega), programerji pa včasih z zamikanjem nakažejo ravno drugače. Poglejmo si primer:

```
if (pogoj1)  
    if (pogoj2) stavek1;  
else stavek2;
```

Stavki so zamaknjeni tako, da se zdi, kot da `else` pripada prvemu pogojnemu stavku, v resnici pa pripada drugemu. Če bi želeli, da `else` pripada prvemu pogojnemu stavku, bi morali uporabiti zavite oklepaje:

```
if (pogoj1) {
```

```

    if (pogoj2) stavek1;
}
else stavek2;

```

S pogojnim stavkom se lahko odločimo za eno izmed dveh možnosti. Včasih pa je možnosti več. V takih primerih lahko uporabimo posebno varianto gnezdenih pogojnih stavkov.

```

if (pogoj1) stavek1;
else if (pogoj2) stavek2;
else if (pogoj3) stavek3;
else if (pogoj4) stavek4;
else stavek5;

```

Primer

Recimo, da imamo podan pravokotnik v ravnini, ki ima stranice vzporedne s koordinatnima osema. Naj bo (x_1, y_1) levo spodnje oglišče, (x_2, y_2) pa desno zgornje oglišče tega pravokotnika. Točka (x, y) naj bo poljubna točka v ravnini. Zanima nas, ali ta točka leži v notranjosti, zunanosti, ali pa na robu pravokotnika.

```

printf("Točka (%d, %d) leži ", x, y);
if (x1 < x && x < x2 && y1 < y && y < y2) printf("v pravokotniku");
else if (x < x1 || x2 < x || y < y1 || y2 < y) printf("zunaj
pravokotnika");
else printf("na robu pravokotnika");
printf(" [%d, %d] x [%d, %d].\n", x1, x2, y1, y2);

```

Primer

Recimo, da imamo podani dve premici v ravnini. Vsaka je podana s smernim koeficientom in začetno vrednostjo, pri čemer k_1 in n_1 določata prvo premico, k_2 in n_2 pa drugo premico. Zanima nas, ali se premici prekrivata, sta vzporedni, ali pa se sekata v eni točki.

```

if (k1 == k2) {
    if (n1 == n2) printf("Premici sta enaki.\n");
    else printf("Premici sta vzporedni.\n");
}
else {
    double x = (n2 - n1) / (k1 - k2);
    double y = k1 * x + n1;
    printf("Premici se seceta v tocki (%g, %g).\n", x, y);
}

```