

# Programski jezik C#

## Pogojni stavek

**Matija Lokar in Srečo Uranič**

V 0.91

oktober 2008



# Predgovor

---

*Omenjeno gradivo predstavlja naslednji (drugi) del gradiv, namenjenih predmetu Programiranje 1 na višješolskem študiju Informatika.*

*Pokriva pogojni stavek v jeziku C#. Glede samega besedila velja tisto, kar je bilo napisano v predgovorih prejšnjih delov gradiv, torej – ne poveva vsega, določene stvari poenostaviva, veliko je zgledov ... .*

*Gradivo vsekakor ni dokončano in predstavlja delovno različico. V njem so zagotovo napake (upava, da čimmanj), za katere se vnaprej opravičujem. Da bo lažje spremljati spremembe, obstaja razdelek Zgodovina sprememb, kamor bova vpisovala spremembe med eno in drugo različico. Tako bo nekemu, ki si je prenesel starejšo različico, lažje ugotoviti, kaj je bilo v novi različici spremenjeno.*

*Matija Lokar in Srečo Uranič*

# Zgodovina sprememb

---

**6. 10. 2008:** Različica V0.8 – prva, ki je na voljo javno

**13. 10. 2008:** Različica V0.9

**19. 10. 2008:** Različica V0.91

# KAZALO

<b>KAZALO</b> .....	<b>5</b>
<b>Logične vrednosti in logične spremenljivke</b> .....	<b>6</b>
<i>Osnovne logične operacije</i> .....	6
<i>Primerjalne operacije</i> .....	7
<b>Pogojni stavki</b> .....	<b>8</b>
<i>Pogojni stavek if</i> .....	8
<i>Zgledi</i> .....	8
Dvig denarja na bankomatu .....	10
Uredi tri števila od najmanjšega do največjega .....	10
<i>If – else</i> .....	12
Večje/manjše .....	13
Praznovanje rojstnega dne .....	13
<i>Gnezdeni stavek if</i> .....	14
Sodo/liho število manjše/večje od 10 .....	14
Telesna teža .....	14
Nakup stanovanja .....	15
Kvadratna enačba $ax^2 + bx + c = 0$ .....	17
Absolutna vrednost.....	18
Pravilni večkotnik.....	18
Presek premic.....	19
Dvomestno število.....	20
Vrsta osebe.....	20
<b>Naloge</b> .....	<b>21</b>
<b>Naloge s sestavljanjem programov</b> .....	<b>23</b>

## Logične vrednosti in logične spremenljivke

Logične spremenljivke so spremenljivke, ki lahko zavzamejo le eno izmed dveh logičnih vrednosti, **true** ali pa **false**. Označimo jih s tipom **bool**. V spremenljivki tipa **bool** torej lahko shranjujemo le vrednost pravilno (**true**) ali pa napačno (**false**).

Primeri:

```
bool trditev = true;
bool konec = false;
```

Poglejmo preprost primer uporabe podatkovnega tipa *bool*.

```
1: class TipBool {
2:     static void Main(string[] args) {
3:         bool da = true;
4:         bool ne = false;
5:
6:         Console.WriteLine(da);
7:         Console.WriteLine(ne);
8:     } // main
9: } // TipBool
```

Program prevedemo in požemo:

```
true
false
```

### Osnovne logične operacije

Logična operacija je operacija med logičnimi vrednostmi. Rezultat logične operacije je logična vrednost **true** (pravilno) ali **false** (napačno). Logične vrednosti lahko združujemo z logičnimi operatorji: **in**, **ali** ter **negacija**. Naj bosta a in b logična izraza. Poglejmo si logične operacije med njima

Logični operator	Pomen	Logična operacija	Pomen	Opis
<b>&amp;&amp;</b>	Logični in ( <b>and</b> ).	a && b	logični in	Vrne true, kadar sta oba pogoja resnična.
<b>  </b>	Logični ali ( <b>or</b> ).	a    b	logični ali	Vrne false, kadar je vsaj en pogoj resničen.
<b>!</b>	Negacija ( <b>not</b> ).	!a	logična negacija	Vrne false, če je a true in true, če je a false.

Če je združenih več pogojev, lahko uporabimo oklepaje. Vrstni red operacij je takšen, kot to velja pri matematiki. Zato se && izvede pred ||. Kadar nismo prepričani, kakšna je prioriteta operatorjev, uporabimo

oklepaje, da zagotovimo vrednotenje v želenem vrstnem redu. Opozoriti velja, da se izrazi, v katerih uporabljamo operatorja `&&` in `||`, vrednotijo vedno od leve proti desni in sicer toliko časa, da je vrednost celega izraza določena.

Naj bosta **a** in **b** dva logična izraza. Če ima izraz **a** vrednost **false**, potem ima tudi izraz **a && b** vrednost **false**. Če ima izraz **a** vrednost **true**, potem ima tudi izraz **a || b** vrednost **true**. Izraza **b** C# v teh dveh primerih sploh ne izračuna.

## Primerjalne operacije

S primerjalnimi operatorji primerjamo dva izraza. Rezultat primerjalnih operatorjev je vedno tipa *bool*. Najpogosteje se z njimi srečujemo v zankah in pogojnih stavkih.

Relacijski operator	Pomen
<	Manjše.
>	Večje.
==	Enako.
<=	Manjše ali enako.
>=	Večje ali enako.
!=	Različno.

Naj bosta **a** in **b** izraza, katerih vrednosti sta števili. Poglejmo si primerjalne operacije med njima

Primerjalna operacija	Pomen	Opis
<b>a == b</b>	enako	Pogoj je resničen, če sta vrednosti obeh izrazov enaki.
<b>a != b</b>	različno	Pogoj je resničen, če sta vrednosti obeh izrazov različni.
<b>a &gt; b</b>	večje	Pogoj je resničen, če je vrednost levega izraza večja od vrednosti desnega izraza.
<b>a &gt;= b</b>	večje ali enako	Pogoj je resničen, če je vrednost levega izraza večja ali enaka od vrednosti desnega izraza.
<b>a &lt; b</b>	manjše	Pogoj je resničen, če je vrednost levega izraza manjša od vrednosti desnega izraza
<b>a &lt;= b</b>	manjše ali enako	Pogoj je resničen, če je vrednost levega izraza manjša ali enaka od vrednosti desnega izraza.

## Pogojni stavki

Vsi stavki, ki smo jih spoznali do sedaj, so se izvajali zaporedoma. Najprej se je izvedel prvi stavek v metodi *main*, nato naslednji in tako naprej. Pogosto pa postopek zahteva, da kak ukaz izvedemo le, če so izpolnjeni določeni pogoji. To nam omogoča pogojni stavek.

### Pogojni stavek if

Pogojni stavek *if* uporabimo, kadar želimo izvesti določene stavke (oz. stavek) samo v primeru, ko je izpolnjen nek pogoj. Za besedo *if* v oklepajih zapišemo *pogoj*. To je poljuben logični izraz. Sledi mu blok stavkov (oz. stavek), ki jih želimo izvesti v primeru, če je pogoj resničen.

```
if(pogoj) {
    stavek1;
    stavek2;
    ...;
    stavekn;
}
```

Če je blok sestavljen samo iz enega stavka, lahko oklepaje izpustimo.

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

Kljub temu je bolje, da zaradi boljše preglednosti zavite oklepaje `{}` ohranimo.

### Zgledi

Ali je prvo število manjše ali enako drugemu številu?

Napišimo preprost program, ki bo primerjal celi števili, ki sta shranjeni v dveh spremenljivkah. V primeru, da je prvo število manjše ali enako drugemu, bo izpisal "vrednost prvega števila" *je manjše ali enako* "od vrednosti drugega števila". Če je prva vrednost 5, druga pa 7, bo program izpisal "5 je manjše ali enako 7."

```
1: class StavekIf1 {
2:     static void Main(string[] args) {
3:         int prvoStevilo = 5;
4:         int drugoStevilo = 7;
5:
6:         if (prvoStevilo <= drugoStevilo) {
7:             Console.WriteLine (prvoStevilo + " je manjše ali " +
8:                                 "enako " + drugoStevilo + ".");
9:         } // if
10:        Console.WriteLine ("Konec programa.");
11:    } // main
12: } // StavekIf1
```

Program prevedemo in požemo:

```
5 je manjše ali enako 7.
Konec programa.
```



V 6. vrstici se preveri pogoj `prvoStevilo <= drugoStevilo`. Ker je pogoj izpolnjen, saj je 5 manjše ali enako 7, se izpišeta vrednosti v spremenljivkah `prvoStevilo` in `drugoStevilo` z ustreznim tekstom. Program se nato nadaljuje v vrstici 10. Izpiše se besedilo *Konec programa..*

Kaj pa se zgodi, če prvemu številu priredimo vrednost 7, drugemu pa 5?

```

1:  class StavekIf2 {
2:      static void Main(string[] args) {
3:          int prvoStevilo = 7;
4:          int drugoStevilo = 5;
5:
6:          if(prvoStevilo <= drugoStevilo) {
7:              Console.WriteLine (prvoStevilo + " je manjše ali " +
8:                                  "enako " + drugoStevilo + ".");
9:          } // if
10:         Console.WriteLine ("Konec programa.");
11:     } // main
12: } // StavekIf2

```

Program prevedemo in požnemo:

```
Konec programa.
```

#### Opis programa.

V 6. vrstici se preveri pogoj `prvoStevilo <= drugoStevilo`. Ker pogoj ni resničen (ima vrednost `false`), se stavek `Console.WriteLine ()` znotraj pogojnega stavka sploh ne izvede in program se nadaljuje za stavkom `if` v vrstici 10. Izpiše se besedilo *Konec programa..*

Ukaz v vrstici za stavkom `if` (10. vrstica v našem primeru) se izvede vedno, ukaz v vrstici 7 in 8 (stavek znotraj stavka `if`) pa le, če je pogoj izpolnjen.

Spremenimo program tako, da bomo prebrali dve števili in izpisali v kakšnem razmerju je prvo število z drugim.

```

3:  class StavekIf3 {
4:      static void Main(string[] args) {
5:          Console.Write("Prvo stevilo: ");
6:          String beri = Console.ReadLine();
7:
8:          int prvoStevilo = int.Parse(beri);
9:          Console.Write("Drugo stevilo:");
10:         int drugoStevilo = int.Parse(Console.ReadLine());
11:
12:         if(prvoStevilo <= drugoStevilo) {
13:             Console.WriteLine (prvoStevilo + " je manjše ali " +
14:                                 "enako " + drugoStevilo + ".");
15:         } // if
16:         if(prvoStevilo > drugoStevilo) {
17:             Console.WriteLine (prvoStevilo + " je večje od " +
18:                                 drugoStevilo + ".");
19:         } // if
20:     } // main
21: } // StavekIf3

```

#### Opis programa.

V 5. in 6. vrstici prikažemo sporočilo, ki uporabniku pove, naj vnese prvo število. Ko uporabnik zapiše niz in pritisne tipko Enter, se zapisani niz prenese v niz `beri`. S klicem metode `int.Parse(beri)` niz `beri`

pretvorimo v celoštevilčno vrednost, ki se nato priredi spremenljivki *prvoStevilo*. V 8. vrstici ponovno prikažemo sporočilo. Ko pri izvajanju 9. vrstice uporabnik zapiše niz in pritisne tipko Enter, vnešeni pretvorimo v celoštevilčno vrednost, ki jo shranimo v spremenljivko *drugoStevilo*. V 11. vrstici preverimo pogoj in če je resničen, se izpiše ustrezen tekst (vrstici 12 in 13). Isto ponovimo v 15. vrstici. Preveri se pogoj in če je resničen, se izpiše ustrezen tekst (vrstici 16 in 17).

Ker je v našem primeru lahko resničen samo en pogoj, bi bilo bolj smiselno uporabiti pogojni stavek v obliki *if-else*. Tega si bomo ogledali v naslednjem razdelku.

### Dvig denarja na bankomatu

Na transakcijskem računu imamo dovoljen limit 500 €. Ker nam je zmanjkalo denarja, želimo dvigniti določen znesek. Če ne presegamo limita, nam bankomat izplača izbrani znesek. Napišimo program, ki bo v primeru opravljene transakcije izpisal dvignjen znesek, stanje na računu in razpoložljivo stanje na računu. Če transakcija ni dovoljena, bo izpisal le stanje in razpoložljivo stanje na računu. Stanje na računu in znesek dviga preberemo.

```

3:     class DvigDenarja {
4:         static void Main(string[] args) {
5:             const int LIMIT = -500; // dovoljen limit
6:             Console.WriteLine("Stanje na racunu: ");
7:             String beri = Console.ReadLine();
8:             int stanje = int.Parse(beri);
9:             Console.WriteLine("Znesek dviga: ");
10:            beri = Console.ReadLine();
11:            int dvig = int.Parse(beri);
12:
13:            if(stanje - dvig >= LIMIT) {
14:                Console.WriteLine ("Dvig: " + dvig + " SIT");
15:                stanje = stanje - dvig;
16:            } // if
17:
18:            int razpolozljivoStanje = stanje + Math.Abs(LIMIT);
19:
20:            Console.WriteLine ("Stanje na racunu: " + stanje +
21:                               " SIT");
22:            Console.WriteLine ("Razpolozljivo stanje: " +
23:                               razpolozljivoStanje + " SIT");
24:        } // main
25:    } // DvigDenarja

```

#### Opis programa.

V vrstici 5 deklariramo konstanto in ji priredimo vrednost. Sledi vnos (branje) podatkov. Ko podatke preberemo in jih pretvorimo v cela števila, se preveri pogoj v vrstici 13. Če je pogoj resničen (*stanje - dvig >= LIMIT*), se izvedeta stavka znotraj *if* (vrstici 14 in 15). V 18. vrstici sledi izračun razpoložljivega stanja. Izračuna se vrednost izraza, ki se priredi spremenljivki *razpolozljivoStanje*. Pomagali smo si z metodo *Abs()* iz razreda *Math*, ki vrne absolutno vrednost števila. V našem primeru je to *10000*. Na koncu še izpišemo stanje na računu ter razpoložljivo stanje.

### Uredi tri števila od najmanjšega do največjega

Napišimo program, ki bo prebral tri števila in jih izpisal v naraščajočem vrstnem redu. Pri tem bomo večkrat morali zamenjati vrednost dveh spremenljivk. To smo se že naučili ob programu *ZamenjavaSpremenljivk*, ki smo ga napisali v razdelku o celih številih. Vrednosti v spremenljivkah *a*, *b* in *c* bomo zamenjali tako, da bo v *a* najmanjša vrednost, v *c* pa največja vrednost. Nato izpišemo števila v vrstnem redu *a*, *b*, *c*.

```
1: class UrediTriStevila {
2:     static void Main(string[] args) {
3:
4:         // preberemo podatke (nize) in jih
5:         // pretvorimo v cela stevila
6:         Console.Write("Prvo število:");
7:         string beri = Console.ReadLine();
8:         int a = int.Parse(beri);
9:         Console.Write("Drugo število:");
10:        beri = Console.ReadLine();
11:        int b = int.Parse(beri);
12:        Console.Write("Tretje število:");
13:        beri = Console.ReadLine();
14:        int c = int.Parse(beri);
15:
16:        Console.WriteLine ("Vnesel si: " + a + " " + b + " " +c);
17:        if(b < a) {
18:            // zamenjaj a in b
19:            int t = b;
20:            b = a;
21:            a = t;
22:        } // if
23:        // sedaj zagotovo velja a <= b
24:        if(c < a) {
25:            // zamenjaj a in c
26:            int t = c;
27:            c = a;
28:            a = t;
29:        } // if
30:
31:        // sedaj je a najmanjše izmed treh števil
32:
33:        if(c < b) {
34:            // zamenjaj b in c
35:            int t = b;
36:            b = c;
37:            c = t;
38:        } // if
39:
40:        // v c je sedaj največje izmed treh števil
41:
42:        Console.WriteLine ("Urejena števila po velikosti " +
43:                            "od najmanjšega do največjega:");
44:        Console.Write (a + " " + b + " " + c);
45:    } // main
46: } // UrediTriStevila
```

### Opis programa.

Preberemo podatke in jih pretvorimo v cela števila. Sledi izpis vrednosti, ki so shranjene v spremenljivkah *a*, *b* in *c*, opremljene z ustreznim tekstom. V 17. vrstici poskrbimo, da je  $a \leq b$ . Preverimo, če to ne drži. Če je vrednost spremenljivke *b* manjša od vrednosti spremenljivke *a*, se njuni vrednosti zamenjata (19.- 21. vrstice).

Sedaj zagotovimo, da je *a* najmanjše število. Od *b* je že manjše (ali kvečjemu enako). Zato preverimo razmerje do spremenljivke *c*. Če je v spremenljivki *c* slučajno manjša vrednost, jo z zamenjavo »preselimo« v *a*. Tako je potem v *a* zagotovo najmanjša vrednost.

V  $a$  je že najmanjše število, ne vemo pa, v kakšnem razmerju sta vrednosti v spremenljivkah  $b$  in  $c$ . S pogojem v vrstici 33 in morebitno zamenjavo smo zagotovili, da je v spremenljivki  $c$  večje od obeh števil in s tem tudi največje število. Vrstni red vrednosti je torej  $a, b$  in  $c$ .

Na koncu še izpišemo urejene vrednosti z ustreznim tekstom.

Če si izberemo števila v padajočem vrstnem redu (npr.  $7, 4$  in  $1$ ), bodo resnični vsi trije stavki *if*. Ker je  $b$  ( $4$ ) manjše od  $a$  ( $7$ ), se izvedejo stavki znotraj stavka *if* in zamenjata se vrednosti spremenljivk  $a$  in  $b$  (vrstice 17 do 22). Sedaj so vrednosti spremenljivk  $a = 4, b = 7, c = 1$ . Ker je  $c$  ( $1$ ) manjše od  $a$  ( $4$ ), se zamenjata ti dve vrednosti (vrstice 24 do 29). Sedaj so vrednosti spremenljivk  $a = 1, b = 7, c = 4$ . Ker je  $c$  ( $4$ ) manjše od  $b$  ( $7$ ), se zamenjata ti dve vrednosti (vrstice 33 do 38). Sedaj so vrednosti spremenljivk  $a = 1, b = 4, c = 7$ . Te nato opremljene z ustreznim tekstom izpišemo (vrstice 42 do 44).

Če si izberemo števila, ki so že urejena v naraščajočem vrstnem redu (npr.  $1, 2$  in  $3$ ), bodo neresnični vsi trije pogoji pri stavkih *if*. Zato se ne bo izvedel noben stavek znotraj stavkov *if*. V primeru omenjenem prej (podatki so npr.  $1, 2, 3$ ) se bo po vrstici 16 preveril pogoj v vrstici 17 in ker je njegova vrednost *false* ( $3$  ni manjše od  $1$ ), se stavki znotraj stavka *if* ne izvedejo. Prav tako se zgodi v vrstici 24 in 33. V tem primeru izpišemo vrednosti v takšnem vrstnem redu, kot smo jih vnesli (vrstice 42 do 44).

## If - else

Prejšnja oblika pogojnega stavka je poskrbela, da se je določen ukaz izvedel le, če je bil pogoj izpolnjen. Nato smo v vsakem primeru nadaljevali z istim ukazom. Včasih pa želimo, da se takrat, ko pogoj ni izpolnjen, zgodi nekaj drugega. Takrat uporabimo stavek *if* v kombinaciji z *else*.

```
if(pogoj) {
    stavekp1;
    stavekp2;
    ...;
    stavekpn;
}
else {
    stavekr1;
    stavekr2;
    ...;
    stavekrk;
}
```

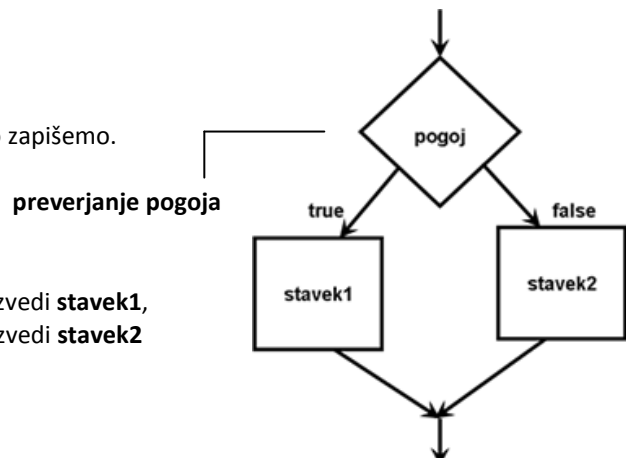
Če bo pogoj izpolnjen, se bodo izvedli  $stavekp_1$  do  $stavekp_n$  (oz. *stavek1* v spodnjem zgledu), sicer  $stavekr_1$  do  $stavekr_k$  (oz. *stavek2* v spodnjem zgledu). Če sta bloka sestavljena samo iz enega stavka, lahko oklepaje izpustimo

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

vendar jih zaradi boljše preglednosti praviloma vseeno zapišemo.

Schematično prikažemo stavek *if-else* takole:

če je resničen, izvedi **stavek1**,  
če ni resničen, izvedi **stavek2**



## Večje/manjše

Spomnimo se programa *StavekIf3*, kjer smo prebrali dve števili in izpisali, v kakšnem razmerju je prvo število z drugim. Popravimo program tako, da bomo namesto dveh zaporednih pogojnih stavkov uporabili stavek *if-else*.

```
3:     class StavekIfElse {
4:         static void Main(string[] args) {
5:             Console.WriteLine("Prvo stevilo: ");
6:             String beri = Console.ReadLine();
7:             int prvoStevilo = int.Parse(beri);
8:             Console.WriteLine("Drugo stevilo:");
9:             int drugoStevilo = int.Parse(Console.ReadLine());
10:
11:            if(prvoStevilo <= drugoStevilo) {
12:                Console.WriteLine (prvoStevilo + " je manjše ali " +
13:                                   "enako " + drugoStevilo + ".");
14:            } else {
15:                Console.WriteLine (prvoStevilo + " je večje od " +
16:                                   drugoStevilo + ".");
17:            } // else
18:            Console.WriteLine ("Konec programa.");
19:        } // main
20:    } // StavekIfElse
```

Popravili smo vrstico 14. Če je pogoj v 11. vrstici resničen (kadar je prvo prebrano število manjše ali enako drugemu prebranemu številu), se izvedeta stavka v vrsticah 12 in 13. Nato nadaljujemo v vrstici 19. V primeru, če pogoj v 11. vrstici ni resničen, je prvo prebrano število večje od drugega prebranega števila. Takrat se izvede stavek v vrstici 16 in stavek v vrstici 17. Nato nadaljujemo v vrstici 19.

## Praznovanje rojstnega dne

Janja ima rojstni dan 29. februarja. Če leto ni prestopno, ga praznuje 1. marca. Napišimo program, ki bo Janji pomagal ugotoviti, ali bo na določeno leto praznovala rojstni dan 29. februarja ali 1. marca.

Leto je prestopno, kadar je deljivo s 4 in ne s 100 ali kadar je deljivo s 400.

```
3:     class Praznovanje {
4:         static void Main(string[] args) {
5:             Console.WriteLine("Vpisi leto: ");
6:             string beri = Console.ReadLine();
7:             int leto = int.Parse(beri);
8:
9:             if (((leto % 4 == 0) && (leto % 100 != 0))
10:                || (leto % 400 == 0)) {
11:                 Console.WriteLine ("Praznovanje bo 29. februarja.");
12:             } // if
13:             else {
14:                 Console.WriteLine ("Praznovanje bo 1. marca.");
15:             } //else
16:         } // main
17:     } // Praznovanje
```

Prebrani niz pretvorimo v celo število(vrstice 5 do 7). Nato preverimo pogoj v vrstici 9 in 10. Pogojni stavek je sestavljen iz dveh delov. Prvi del `((leto % 4 == 0) && (leto % 100 != 0))` je sestavljen iz dveh

pod pogojev. Če sta pod pogoja resnična, se drugi del sploh ne preverja in izvede se vrstica 11, ki izpiše "Praznovanje bo 29. februarja.". V primeru, da prvi del ni resničen, se preveri drugi del. Če je resničen drugi del, se prav tako izpiše "Praznovanje bo 29. februarja.". Če pa sta oba neresnična, se izvede else del (vrstica 13). Izpiše se "Praznovanje bo 1. marca." (vrstica 14).

## Gnezdeni stavek if

Gnezdeni stavek **if** je stavek **if** znotraj stavka **if**. Pri takih stavkih moramo biti pozorni na to, v katerem primeru se izvede veja **else** takega gnezdenega stavka. Pomagamo si seveda z oklepaji **{}**.

Pogosto imamo opraviti tudi s kombinacijo več pogojnih stavkov **if...else**. Znotraj pogojnega stavka imamo spet pogojni stavek. Kombinacijo običajno zapišemo v obliki

```
if (pogoj1) {
    stavek1;
}
else if (pogoj2) {
    stavek2;
}
...
else {
    stavekn;
}
```

Če velja *pogoj1*, se izvede *stavek1*, sicer če velja *pogoj2*, se izvede *stavek2*, sicer ..., sicer se izvede *stavekn*. V navedenem primeru se vedno izvede natanko en od stavkov *stavek1*, *stavek2*, ..., *stavekn*.

## Sodo/liho število manjše/večje od 10

Preberimo celo število in ugotovimo, ali je sodo ali liho število. Pr item nas zanima še, če je manjše ali večje (ali enako) od 10. Izpis naj bo oblike *Sodo število večje od 10*

```
Console.WriteLine("Vnesi celo število: ");
int stevilo = int.Parse(Console.ReadLine());
if (stevilo > 10)
{
    if (stevilo % 2 == 0)
        Console.WriteLine("Sodo število večje od 10");
    else
        Console.WriteLine("Liho število večje od 10");
}
else
{
    if (stevilo % 2 == 0)
        Console.WriteLine("Sodo število manjše od 10");
    else
        Console.WriteLine("Liho število manjše od 10");
}
```

## Telesna teža

Jelka je v lekarni dobila brošuro, v kateri je formula za izračun indeksa telesne mase. Indeks telesne mase je razmerje med telesno maso (v kg) in kvadratom višine (v m). S tem indeksom ugotovimo, če imamo čezmerno

telesno maso. Napišimo program, ki bo prebral podatke o teži in višini in glede na izračunani indeks telesne mase (ITM) izpisal ustrezen tekst.

Če je indeks telesne mase manj kot 18.5, naj program izpiše "Premajhna telesna masa.". V primeru, da je indeks telesne mase med 18.5 in 24.9, naj se izpiše "Normalna telesna masa.". Če pa je indeks telesne mase več kot 25, naj program izpiše "Cezmerna telesna masa.". Na koncu v oklepaju še izpišimo izračunani ITM.

```

2:     class TelesnaTeza {
3:         static void Main(string[] args) {
4:             double itm;
5:             Console.WriteLine("Vnesi maso (v kg): ");
6:             string beri = Console.ReadLine();
7:             int masa = int.Parse(beri);
8:             Console.WriteLine("Vnesi visino (v m):");
9:             beri = Console.ReadLine();
10:            double visina = double.Parse(beri);
11:
12:            // izracunamo itm
13:            itm = masa / visina / visina;
14:
15:            if(itm < 18.5) {
16:                Console.WriteLine ("Premajhna telesna teza." +
17:                                   " (itm = " + itm + ")");
18:            } // if
19:            else if(itm <= 24.9) { // vemo da velja itm >= 18.5
20:                Console.WriteLine ("Normalna telesna teza." +
21:                                   " (itm = " + itm + ")");
22:            } // else if
23:            else {
24:                Console.WriteLine ("Cezmerna telesna teza." +
25:                                   " (itm = " + itm + ")");
26:            } // else
27:        } // main
28:    } // TelesnaTeza

```

### Opis programa.

V vrsticah 15 – 26 imamo primer gnezdenega stavka. Najprej se preveri prvi pogoj. Če je ITM manjši od 18.5 se izpiše *Premajhna telesna teža* in program se zaključi. Če to ni res (ITM je večji ali enak 18.5, preverimo drugi pogoj. Če je naš ITM manjši ali enak 24.9 (večji ali enak 18.5 je že!), se izpiše *Normalna telesna teža*, če pa ta pogoj ni izpolnjen, se izpiše *Cezmerna telesna teža*.

### Nakup stanovanja

Janez se je odločil, da gre živeti na svoje in si začel iskati stanovanje. V časopisu je zasledil nekaj primernih stanovanj. Na banki ima že nekaj privarčevanega denarja. Na banki je zaprosil za informativni izračun kredita. Pomagajmo Janezu pri nakupu. Poznamo ceno stanovanja, znesek privarčevanega denarja ter znesek odobrenega kredita. Napišimo torej program, ki bo glede na vnesene zneske izpisal ustrezen tekst:

Če bo potrebno vzeti kredit, izpišemo, koliko kredita moramo vzeti.

Če ima dovolj privarčevanega denarja (kredita ne bomo vzeli), izpišemo koliko denarja mu ostane po nakupu stanovanja.

Če kljub kreditu nima dovolj denarja, izpišemo, koliko kredita lahko vzame ter koliko mu zmanjka za nakup stanovanja.

V vseh primerih izpišemo, koliko stane stanovanje, koliko denarja je privarčeval ter ali lahko kupi stanovanje.

```
1: // kupovanje stanovanja na kredit
2: class NakupStanovanja {
3:     static void Main(string[] args) {
4:         int privarcevano, cena, kredit, razlika;
5:         string stanje = "Stanovanje lahko kupimo.";
6:
7:         // preberemo zneske
8:         Console.Write("Vnesi ceno stanovanja:");
9:         string beri = Console.ReadLine();
10:        cena = int.Parse(beri);
11:        Console.Write("Vnesi privarcevani znesek: ");
12:        privarcevano = int.Parse(beri);
13:        beri = Console.ReadLine();
14:        cena = int.Parse(beri);
15:        Console.Write("Koliko kredita lahko dobis: ");
16:        beri = Console.ReadLine();
17:        kredit = int.Parse(beri);
18:
19:        // izracunamo razliko
20:        razlika = cena - privarcevano;
21:
22:        Console.WriteLine ("Stanovanje stane " + cena + " SIT," +
23:                            " privarcevano imamo "+ privarcevano +
24:                            " SIT.");
25:
26:        // glede na vrednosti izpisemo ustrezen tekst
27:        if(razlika <= 0) {
28:            Console.WriteLine ("Za nakup stanovanja imamo dovolj" +
29:                                " privarcevanega denarja.");
30:            if(razlika < 0) {
31:                Console.WriteLine ("Ostane nam se " +
32:                                    Math.Abs(razlika) + " SIT.");
33:            } // if
34:            else {
35:                Console.WriteLine ("Porabimo ves privarcevani denar.");
36:            } // else
37:        } // if
38:        else {
39:            Console.WriteLine ("Za nakup stanovanja nimamo dovolj"+
40:                                " privarcevanega denarja.");
41:            if(razlika <= kredit) {
42:                Console.WriteLine ("Kredita moramo vzeti " + razlika
43:                                    + " SIT.");
44:            } // if
45:            else {
46:                Console.Write ("Ker lahko kredita vzamemo le " +
47:                                kredit + " SIT, nam za nakup " +
48:                                "stanovanja zmanjka " +
49:                                (razlika - kredit) + " SIT.");
50:                stanje = "Stanovanja si ne moremo privosciti.";
51:            } // else
52:        } // else
53:        Console.WriteLine (stanje);
54:    } // main
55: } // NakupStanovanja
```

V zgledu vidimo, kako lahko tudi znotraj pogojnega stavka spet uporabimo pogojni stavek. Pri pisanju ustreznih pogojev moramo paziti pri kakšnih pogojih se najdemo znotraj pogojnega stavka – bodisi v pritrdilnem bodisi v delu else.



### Kvadratna enačba $ax^2 + bx + c = 0$

Napišimo program, ki bo poiskal rešitve kvadratne enačbe  $ax^2 + bx + c = 0$ . Pri tem upoštevajmo vse možnosti, ki lahko nastopijo. Po branju koeficientov  $a$ ,  $b$  in  $c$  najprej preverimo, ali je koeficient pri kvadratnem členu enak 0. Če je, imamo opraviti z linearno enačbo. Takrat preverimo še  $b$ . Če je tudi ta 0, sta zopet dve možnosti. Prva, ko je še  $c$  enak 0, pomeni, da imamo opraviti z identiteto, druga pa, da je enačba protislovna. Kadar je  $b$  od 0 različen, imamo eno rešitev, in sicer  $-c / b$ .

Če je  $a$  od 0 različen, imamo opraviti s pravo kvadratno enačbo. Izračunamo njeno diskriminanto  $b^2 - 4ac$ . Če je pozitivna, imamo realne rešitve, do katerih si pomagamo z znanim obrazcem in Vietovo formulo. Če je negativna, imamo kompleksni rešitvi. Pri teh izračunamo realni del in nato še imaginarnega.

```
3:     class KvadratnaEnacba {
4:         static void Main(string[] args) {
5:             double d, x1, x2;
6:
7:             // preberemo vrednosti
8:             Console.WriteLine("Vnesi a: ");
10:            double a = double.Parse(Console.ReadLine());
11:            Console.WriteLine("Vnesi b:");
12:            double b = double.Parse(Console.ReadLine());
13:            Console.WriteLine("Vnesi c:");
14:            double c = double.Parse(Console.ReadLine());
15:
16:            // glede na vrednosti izpisemo ustrezen tekst
17:            if(a == 0) {
18:                if(b == 0) {
19:                    if(c == 0) {
20:                        Console.WriteLine("Enačbo reši vsako " +
21:                            "(kompleksno) število.");
22:                    } // if
23:                } else {
24:                    Console.WriteLine("Enačba nima rešitve.");
25:                } // else
26:            } // if
27:            else {
28:                Console.WriteLine("Edina rešitev enačbe je " +
29:                    "(-c / b) + ".");
30:            } // else
31:        } // if
32:        else {
33:            d = b * b - 4 * a * c;
34:            if(d == 0) {
35:                Console.WriteLine("Enačba ima dvojno ničlo " +
36:                    "(-b / (2 * a)) + ".");
37:            } // if
38:            else if(d > 0) {
39:                if(b > 0) {
40:                    x1 = (-b - Math.sqrt(d)) / (2 * a);
41:                } // if
42:            } else {
43:                x1 = (-b + Math.sqrt(d)) / (2 * a);
44:            } // else
45:            Console.WriteLine("Enačba ima dve realni rešitvi.");
46:            Console.WriteLine("x1 = " + x1 + ".");
47:            Console.WriteLine("x2 = " + (c / a / x1) + ".");
48:        } // else if
49:        else {
50:            x1 = -b / (2 * a);
```

```

51:         x2 = (Math.sqrt(-d)) / (2 * a);
52:         Console.WriteLine("Enačba ima kompleksni rešitvi.");
53:         Console.WriteLine("x1 = " + x1 + " + " +
54:             x2 + " * i.");
55:         Console.WriteLine("x2 = " + x1 + " - " +
56:             x2 + " * i.");
57:     } // else
58: } // else
59: } // main
60: } // KvadratnaEnacba

```

### Absolutna vrednost

Napišimo preprost program, ki bo izpisal absolutno vrednost danega števila. V standardni knjižnici Math sicer obstaja metoda Abs, vendar rajši uporabimo postopek, kjer v primeru, da je dano število negativno, uporabimo nasprotno vrednost tega števila.

```

static void Main(string[] args)
{
    Console.Write("Vnesi število: ");
    double stevilo = double.Parse(Console.ReadLine()); // Preberemo število

    // Preverimo predznak
    if (stevilo < 0)
        stevilo = -stevilo;
    Console.WriteLine("Absolutna vrednost: " + stevilo); // Izpis
}

```

#### Zapis na zaslonu:

```

Vnesi število: -19.5
Absolutna vrednost: 19.5

```

**Razlaga.** Preberemo podatek in ga pretvorimo v realno število. Če je število negativno, določimo nasprotno vrednost tega števila. Na koncu izpišemo absolutno vrednost vnesenega števila.

### Pravilni večkotnik

Sestavimo program, v katerega preko tipkovnice vnesemo število stranic pravilnega (enakostraničnega) večkotnika in dolžino stranice. Program izračuna in izpiše njegov obseg. Če je število stranic premajhno, da bi lahko tvorile lik, izpiše: "Napaka! Število stranic je premajhno."

```

class Veckotnik
{
    static void Main(string[] args)
    {
        // Vneseno število vseh stranic
        Console.Write("Vnesi število stranic večkotnika: ");
        int stranice = int.Parse(Console.ReadLine());

        // Preverimo število stranic
        if (stranice < 3)
        {
            Console.WriteLine("Napaka! Število stranic je premajhno.");
        }
        else

```

```

    {
        // Vnesena dolžina stranic
        Console.WriteLine("Vnesi dolzino stranice: ");
        double dolzina = double.Parse(Console.ReadLine());

        // Izračun in izpis obsega večkotnika
        double obseg = stranice * dolzina;
        Console.WriteLine("Obseg veckotnika je " + obseg + ".");
    }
}

```

**Zapis na zaslonu:**

```

Unesi stevilo stranic veckotnika: 5
Unesi dolzino stranice: 5,3
Obseg veckotnika je 26,5.

```

**Razlaga.** V spremenljivko `stranice` se shrani vneseno število. Če je število manjše od 3, se izpiše "Napaka! Število stranic je premajhno.". Če pa je število večje ali enako 3, se v spremenljivko `dolzina` shrani (novo) vneseno število. Nato se izračuna obseg večkotnika, ki se shrani v spremenljivko `obseg`. Vrednost spremenljivke `obseg` se na koncu izpiše.

**Presek premic**

Napišimo program, ki bo ugotovil, ali se dani dve premici v ravnini  $y = k_1x + n_1$  in  $y = k_2x + n_2$  sekata ali ne. Če se sekata, naj program določi še koordinate preseka.

Tu bomo uporabili gnezdene pogojne stavke. Stavki v telesu pogojnega stavka so namreč poljubni stavki. Torej lahko med njimi spet uporabimo pogojni stavek. Ker je ta znotraj drugega pogojnega stavka ("v gnezdu"), rečemo, da gre za gnezdeni pogojni stavek.

Kadar imata premici enaka smerna koeficienta, sta vzporedni. Če imata enaka tudi prosta člena, se premici pokrivata. Če pa sta smerna koeficienta različna, se premici sekata in sicer v natanko eni točki.

```

class PresekPremic
{
    static void Main(string[] args)
    {
        // Preberemo vrednosti prve premice
        Console.WriteLine("Prva premica: ");
        Console.Write("- smerni koeficient: ");
        double k1 = double.Parse(Console.ReadLine());
        Console.Write("- prosti člen: ");
        double n1 = double.Parse(Console.ReadLine());

        // Preberemo vrednosti druge premice
        Console.WriteLine("Druga premica: ");
        Console.Write("- smerni koeficient: ");
        double k2 = double.Parse(Console.ReadLine());
        Console.Write("- prosti člen: ");
        double n2 = double.Parse(Console.ReadLine());

        Console.WriteLine(); // Za lepši videz
        // Glede na vrednosti izpišemo ustrezen tekst
        if (k1 == k2)
        {
            if (n1 == n2)

```

```

    {
        Console.WriteLine("Premici se pokrivata.");
    }
    else
    {
        Console.WriteLine("Premici sta vzporedni.");
    }
}
else
{
    double x = (n2 - n1) / (k1 - k2);
    double y = k2 * x + n2;

    Console.Write("Premici se seceta v tocki ");
    Console.WriteLine("(" + x + "," + y + ").");
}
}
}

```

### Dvomestno število

Prebrati moramo dvomestno število in izpisati njegove desetice in enice. Če vnešeno število ni dvomestno, naj se izpiše ustrezno obvestilo. Napišimo samo "glavni" del programa

```

Console.Write("Vnesi število : ");
int stevilo=Convert.ToInt32(Console.ReadLine());
if (10 <= stevilo && stevilo <= 100)
{
    Console.WriteLine("desetice : " + stevilo / 10);
    Console.WriteLine("enice : " + stevilo % 10);
} else
    Console.WriteLine("Število ni v dogovorjenih mejah!");

```

### Vrsta osebe

Program naj zahteva vnos starosti neke osebe, nato pa naj izpiše za kakšno vrsto osebe gre in sicer

do 2 leti	Dojenček
3-10 let	Mladoletnik
11-19 let	Najstnik
20 in več	Odrasla oseba

```

C1: Console.Write("Starost osebe v letih: ");
C2: int starost=Convert.ToInt32(Console.ReadLine());
C3: if (starost<=2)
C4:     Console.WriteLine("Dojenček!");
C5: else if (starost <= 10)
C6:     Console.WriteLine("Mladoletnik!");
C7: else if (11 <= starost && starost <= 19)
C8:     Console.WriteLine("Najstnik!");
C9: else Console.WriteLine("Odrasla oseba!");

```

#### Opis:

Oglejmo si le pogoja v vrsticah C5 in C7. Namenoma smo ju napisali na različen način. Namreč tudi v C5 bi lahko napisali pogoju tako, kot smo to naredili v C7, torej (3 <= starost && starost <= 10), a je prvi del

pogoja odveč. Ker se preverjanje tega pogoja izvede le, če pogoj ( $\text{starost} \leq 2$ ) ni izpolnjen, že vemo, da je del  $3 \leq \text{starost}$  resničen, torej ga ni potrebno preverjati!

Prav tako se zaradi preglednosti pogoje splača pisati na "smiselen" način. Tako pogoj raje zapišemo v obliki

```
(11 <= starost && starost <= 19)
```

kot pa denimo

```
(starost >= 11 && starost <= 19)
```

Oba zapisa pogoja sta sicer povsem enakovredna, a verjetno se boste strinjali, da je prvi preglednejši in nam hitreje "pove", da nas zanima, če je starost v mejah med 11 in 19.

## Naloge

1. Ugotovi pravilnost oz. nepravilnost naslednjega pogoja, če je  $x = 5$

```
((3 > x) || (5 <= x)) && (x! = 8)
```

2. Kakšna je vrednost spremenljivke N po izvedbi naslednjega stavka if?

```
int N = 1;
bool B = true;
if ((N < 5) && B)
{
    N = N + 1;
}
else
{
    N = 0;
}
```

3. Kakšno vrednost ima spremenljivka potem, ko se izvede prireditveni stavek:

a) <code>bool f = !(3 &lt;= 5 &amp;&amp; 6 != 7);</code>	f	_____
b) <code>bool g = (3 &gt; 2)    (3 &gt; 5 &amp;&amp; 6 &gt; 3);</code>	g	_____
c) <code>bool b = 12 &lt;= 4 * 3;</code>	b	_____
d) <code>b = true &amp;&amp; (2 &lt; 1);</code>	b	_____
e) <code>b = (7 &lt; 4)    (4 &gt;= 5);</code>	b	_____
f) <code>bool n = !(5 &lt; 5) &amp;&amp; (2 &lt;= 3);</code>	n	_____

4. Katere kombinacije znakov izmed navedenih so relacijski operatorji v C#

- a. `<`
- b. `!=`
- c. `>=`
- d. `<>`
- e. `!=`
- f. `>`
- g. `=`

5. Sestavi logični izraz, ki vrne true, če je spremenljivka x sodo število večje od 50 in manjše od 100, ki ni deljivo s sedem.
6. Sestavi logični izraz, ki ima vrednost true, če je spremenljivka x sodo število med (vključno) 50 in 100 ali pa lihi večkratnik števila 3.
7. Sestavi logični izraz, ki pove, če je celo število v spremenljivki `jazSemStevka` res enomestno število.
8. Napišite logični izraz, ki ima vrednost true takrat, če je pozitivno celo število v spremenljivki `semTromestno` res tromestno in false sicer. 123 je tromestno, števila 23, 1245, 1123 pa niso.
9. Napišite logični izraz, ki ima vrednost true takrat, če je vrednost spremenljivke a, v kateri hranimo decimalna števila, različna od vrednosti spremenljivke b, v kateri prav tako hranimo decimalna števila.
10. Za vsako vrstico spodnjega dela programa izpiši, kakšne vrednosti imajo po izvedbi vrstice tiste spremenljivke, ki nastopajo v njej! Vpisuj le v tiste vrstice, ki se dejansko izvedejo!

	a	b	c	d
<code>int a = 10, b = 3, c, d = 4;</code>				
<code>a = a + b - (int)3.5*d;</code>				
<code>c = a % b;</code>				
<code>if( b % 2 == 0) {</code>				
<code>b = c;</code>				
<code>c = b;</code>				
<code>}</code>				
<code>else {</code>				
<code>c = d;</code>				
<code>d = c;</code>				
<code>}</code>				
<code>a = a + b - d;</code>				
<code>b = c * d - a;</code>				
<code>c = -c;</code>				
<code>d = d + 1;</code>				

11. Za vsako vrstico spodnjega dela programa izpiši, kakšne vrednosti imajo po izvedbi vrstice tiste spremenljivke, ki nastopajo v njej! Vpisuj le v tiste vrstice, ki se dejansko izvedejo!

	a	b	x	z
<code>int a = 42;</code>				
<code>int b = 27;</code>				
<code>int x, z;</code>				
<code>if ( a &lt; b)</code>				
<code>{ x = a;</code>				
<code>a = b;</code>				
<code>b = x;</code>				
<code>}</code>				
<code>else</code>				
<code>{ a = -b;</code>				

b = -a;				
}				
z = a + b;				

## Naloge s sestavljanjem programov

1. Jure je napisal program, ki ima nekaj napak. Poišči jih in jih odpravi. Ugotovi, kaj program dela!

```
class Pogojni {
    static void Main(string[] argc) {
        Write("Vnesi temp: ");
        string beri = Console.preberiPodatek();
        int x = Integer.parseInt(beri);
        if(x < 0) {
            Console.println("Zunaj je mrzlo");
        } else {
            System.WriteLine("Zunaj je toplo");
        }
    }
}
```

2. Dopolni Juretov program tako, da bo izpisal:
  - o **mrzlo**, če je vrednost spremenljivke x manjša od 0,
  - o **hladno**, če je vrednost spremenljivke x med 0 in 15 in
  - o **toplo**, če je vrednost spremenljivke x večja od 15.
3. V Butalah je prava panika. 1. januarja bodo butalske cekine zamenjali za novo valuto, še vedno pa ni znano, kakšen bo menjalni tečaj. No, panika je odveč, saj so iz Banke Butale sporočili, da so vladni modreci odločili, da bo menjalni tečaj odvisen od obnašanja Šprince Marogaste, glavne butalske uši, opoldne na Slivestrovo. Tako bodo imeli celo popoldne in celo noč, da bodo lahko pripravili nove cenike. Zato bo glavni butalski informatik pripravil program, ki bo takoj, ko bo znan menjalni tečaj, za neko ceno izpisal njihovo vrednost v novi valuti.

Ampak razbojnik Cefizelj je ravno tiste dni ukradel edino miško v Butalah. Zato pomagaj Butalcem in ti sestavi ustrezni program. Potrebno bo prebrati staro ceno in menjalni tečaj bo potrebno prebrati. Pri tem upoštevaj, da bo menjalni tečaj dan na decimalke, a nova valuta pozna le cele vrednosti. A pozor, Butalci imajo malo drugačno zaokrožanje – če so desetinke sode, se zaokroži navzdol, če pa lihe, pa navzgor.

4. Sestavi program, ki prebere dve vrednosti, ki predstavljata dolžino in širino oglasne table in nato izpiše velikost največjega reklamnega napisa, ki ga lahko na tablo prilepimo vodoravno in ima stranice v razmerju 2(dolžina):1(širina).

Primeri:

- Če sta podatka 24 in 20 naj program izpiše: Velikost napisa je 24 x 12.
- Če sta podatka 200 in 400 naj program izpiše: Velikost napisa je 200 x 100.
- Če sta podatka 300 in 100 naj program izpiše: Velikost napisa je 200 x 100.
- Če sta podatka 500 in 200 naj program izpiše: Velikost napisa je 400 x 200.
- Če sta podatka 500 in 260 naj program izpiše: Velikost napisa je 500 x 250.

5. Preberi dve celi števili in preveri, če manjše deli večjega. V tem primeru izpiši "#vecjeStevilo je mnogokratnik #manjsega", sicer pa "#manjseStevilo" ne deli #vecjega".

6. Teta Minka se še kar ni navadila na evre. Še vedno vse preračunava v tolarje. Zato napiši program Evri, ki dan znesek v evrih (podan kot decimalno število) pretvori v tolarje (to količino vrne kot rezultat). Če si slučajno že pozabil: 1€ je 239,64 tolarja.
7. Preberi dve dolžini v metrih, decimetrih in centimetrih in ugotovi, katera je večja. Namig: če boš obe dolžini pretvoril v centimetre, bo odločanje veliko lažje!
8. V vojski je Polde od svojega oficirja velikokrat dobil nemogoče naloge, kot je na primer "sestavi trikotnik iz teh treh palic dolgih 10, 2 in 3 metre!". Ker to seveda ni mogoče, je Polde včasih že obupaval, saj mu oficir kljub enournem prepričevanju dostikrat ni verjel, da naloga ni izvedljiva. Ker oficir brezpogojno zaupa računalnikom, Poldetu pomagaj in napiši program, ki prebere tri realna števila in preveri, če je iz njih moč sestaviti trikotnik. Če malo pomislimo, je trikotnik mogoče sestaviti, če je vsaka palica krajša od skupne dolžine ostalih dveh.
9. Vrtnar Polde ima že od mladih nog rad številko 9. Pravzaprav jo ljubi tako zelo, da doživi napad evforije vsakič, ko naleti na Njen večkratnik.

V imenu lastnika posestva, na katerem dela Polde, morate poskrbeti, da zaposleni svoje delo opravijo čim hitreje. Prišla je jesen in po travniku je nastlana debela odeja odpadlega listja. Polde bo svoje delo opravil z največjim užitek, če bo le število listov na travniku deljivo z devet.

Napiši program, v katerega vnesete število listov, on pa vam izpiše "Polde je vesel!", če je to število deljivo z devet in "Polde ga gre raje pit", če število ni deljivo z 9.

10. Smolčki praznujejo rojstni dan 29. februarja. Če leto ni prestopno, ga praznujejo 1. marca. Napišite program, ki bo smolčkom sporočil, ali bodo na določeno leto praznovali rojstni dan 29. februarja ali 1. marca.  
**Namig:** Leto je prestopno, kadar je deljivo s 4 in ni deljivo s 100 ali kadar je deljivo s 400.

11. Napišite program, ki zahteva vnos stranic trikotnika in ugotovi, ali tak trikotnik obstaja. Če obstaja, ugotovite, če:
  - je trikotnik pravokoten
  - je trikotnik enakokrak
  - je trikotnik enakostraničen.

**Namig:** Pogoj, da lahko tri števila določajo stranice trikotnika je ta, da so stranice pozitivna števila, posamezna stranica pa mora biti manjša kot vsota drugih dveh.

12. Sestavite program, ki bo čim lepše izpisal vsoto dveh danih kompleksnih števil s celimi komponentami, na primer:

$$\begin{array}{r} 5 \quad + \quad -2 + i = 3 + i \\ 1 + 3i + \quad -1 - 2i = \quad i \\ 2 + i + \quad - i = 2 \end{array}$$

13. Sestavite program, ki bo prebrano oceno po ameriški lestvici (torej znak A, B, ...) pretvoril v slovensko. Pri tem upoštevajte, da velja:

- *A odlično,*
- *B prav dobro,*
- *C in D dobro,*
- *E zadostno in*
- *F nezadostno.*

Oceno izpišite na zaslon z besedo.



14. V spremenljivki *karta* je število od 0 do 51, ki predstavlja eno od igralnih kart za poker. Karte so predstavljene s celoštevilskimi vrednostmi od 0 do 51 kot prikazuje tabela. Napišite program, ki bo na podlagi števila izpisal vrednost in barvo karte.

**Namig:**  $52 = 13 \times 4$

Barva/karta	2	3	4	5	6	7	8	9	10	J	Q	K	A
križ	0	1	2	3	4	5	6	7	8	9	10	11	12
kara	13	14	15	16	17	18	19	20	21	22	23	24	25
srce	26	27	28	29	30	31	32	33	34	35	36	37	38
pik	39	40	41	42	43	44	45	46	47	48	49	50	51

15. Napišite program, ki zna izračunati obseg in ploščino geometrijskih likov. Uporabniku ponudite nabor geometrijskih likov (1 – kvadrat, 2 – pravokotnik, 3 – pravokotni trikotnik in 4 - krog). Ugotovite uporabnikovo izbiro, nato pa v odvisnosti od izbranega lika zahtevajte vnos ustreznih podatkov. Nato izpišite obseg in ploščino lika.
16. Napiši program, ki zahteva vnos stranic trikotnika in ugotovi, ali tak trikotnik sploh obstaja, ali je trikotnik pravokoten, ali je trikotnik enakokrak in ali je trikotnik mogoče enakostraničen.
17. Napišim program, ki bo pomagal ugotoviti, ali bo na določeno leto prestopno ali ne. Leto je prestopno, kadar je deljivo s 4 in ne s 100 ali kadar je deljivo s 400.
18. Preberi poljubno celo število. Ugotovi in izpiši, ali je sodo ali liho!
19. Šola organizira izlet za učence na šoli. Cena izleta je 450 €, če se prijavi do vključno 30 učencev. Vsaka naslednja prijava zniža ceno na posameznika za 1%. Če se prijavi manj kot 20 učencev izlet ne bo organiziran. Napiši program, ki bo zahteval vnos števila učencev in nato izračunal in izpisal ceno izleta na posameznega učenca, oziroma, da je premalo prijav.
20. Sestavi program, ki ugotovi, ali je poljubno trimestno število deljivo s svojo srednjo cifro.
21. Indeks staranja prebivalstva izračunamo tako, da število prebivalcev starejših od 60 let delimo s številom mladih do 20 in količnik pomnožimo s 100. Na podlagi rezultatov lahko ugotovimo tip starostne strukture. Napiši program, ki bo prebral število mladih do 20 let in starejših od 60 let. Na podlagi teh podatkov izračunaj indeks staranja in s pomočjo pogojnih stavkov določi starostni tip prebivalstva ter ga izpiši.

22. Realiziraj naslednjo funkcijo :

$$f(x) = \begin{cases} -3 & ; x < -3 \\ -3*x+2 & ; -3 \leq x < 0 \\ 7 & ; x = 0 \\ 3*x - 2 & ; 0 < x < 4 \\ 3 & ; x \geq 4 \end{cases}$$

23. Program naj prebere poljubno cifro in jo izpiše z besedo. V primeru napačnega vnosa naj se izpiše ustrezno obvestilo.
24. Napiši program, ki prebere tromestno celo število in ugotovi, ali je število desetice večje od vsote enic in stotic tega števila.

Primeri:

*Vpiši število: 234;*  
*V 234 desetice NISO večje kot enice + stotice.*

*Vpiši število: -164;*  
*V -154 desetice SO večje kot enice + stotice.*

Vpiši število: -738;  
V -738 desetice NISO večje kot enice + stotice.

Vpiši število: 494;  
V 404 desetice SO večje kot enice + stotice.

25. Napiši program, ki pretvarja poljuben znesek podan v EUR v drugo valuto, npr. USD in obratno. Na ekranu naj bo sporočilo:

*Pretvarjanje v drugo valuto:*  
1 – EUR v USD, 2 – USD v EUR  
Izberi: \_

Po uporabnikovi odločitvi zahtevaj vnos zneska za menjavo in ustrezen menjalniški tečaj, nato pa s izračunaj in izpiši rezultat.

12. Prodajalec limon bi rad čimprej prodal svojo robo. Vsakih nekaj minut se oglasi s klicem KUPITE! KUPITE ŠE 3 LIMONE! oziroma KUPITE! KUPITE ŠE 75 LIMON!, glede na to, koliko limon mu je še ostalo. Seveda je po nekaj dneh že čisto hripav. Zato mu boste pripravili sintetizator govora, ki bo vpil namesto njega.

Za začetek napišite program, ki bo prebral število limon in izpisal stavek KUPITE! KUPITE ŠE x LIMON! in to v pravilni slovenščini (še 1 limono, še 2 limoni, še 3 limone,...).

- namig1: ločimo primere:
  - \* 1 (limono)
  - \* 2 (limoni)
  - \* 3 (limone)
  - \* 4 (limone)
  - \* 5 ali več (limon)
- namig2: zgodba se ponovi na vsakih 100 (dejansko za pravilno obliko upoštevamo ostanek pri deljenju s 100)