

Struktura grananja

Srednja škola fra Slavka Barbarića Čitluk
Izborna nastava

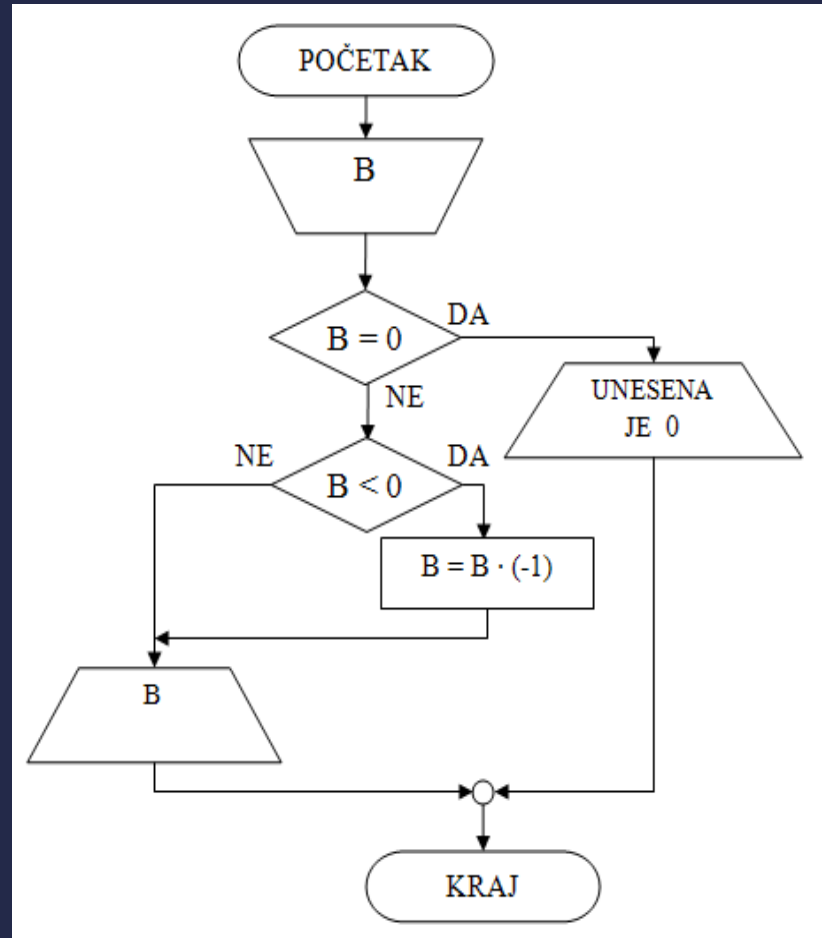
Mario Miletić

Programska struktura grananja

- Za rješavanje većine zadataka potrebne su programske strukture kod kojih **redoslijed izvršavanja naredbi** ovisi o **vrijednostima podataka koji se obrađuju**.
- Grananje je programska struktura koja omogućuje **različit tijek programa, ovisno o rezultatu postavljenog uvjeta**.

Programska struktura grananja

- Primjer programske strukture grananja:
- Korisnik unosi cijeli broj, a zatim se računa apsolutna vrijednost tog broja i ispisuje rezultat.



Blok naredbi

- Dijelovi programa koji se uvjetno izvode grupiraju se u blokove naredbi.
- Blok naredbi se omeđuje parom vitičastih zagrada, zbog preglednosti piše se uvučeno.

```
{  
  int a, b;  
  float c;  
  printf("Unesi prvi broj:");  
  scanf("%d", &a);  
}
```

Zagrade se mogu izostaviti ako se blok sastoji od jedne naredbe.

Lokalne varijable

- Varijable deklarirane unutar bloka naredbi nazivaju se lokalne varijable.
- Ako se varijable deklariraju unutar bloka, postoje samo unutar bloka u kome su deklarirane, u glavnoj funkciji one ne postoje.

Primjer 1

- Primjer lokalne varijable:
 - Treba deklarirati cjelobrojnu varijablu `a` unutar zasebnog bloka, pa joj pridružiti vrijednost. Pokušati ispisati vrijednost varijable `a` u glavnoj funkciji (izvan bloka).

```
Upisi cijeli broj:  
Vrijednost varijable iz bloka a=....
```

Primjer 1

■ Rješenje

```
#include<stdio.h>

void main(void) {

    {
    int a;
    printf("Upiši cijeli broj:");
    scanf ("%d", &a);
    }

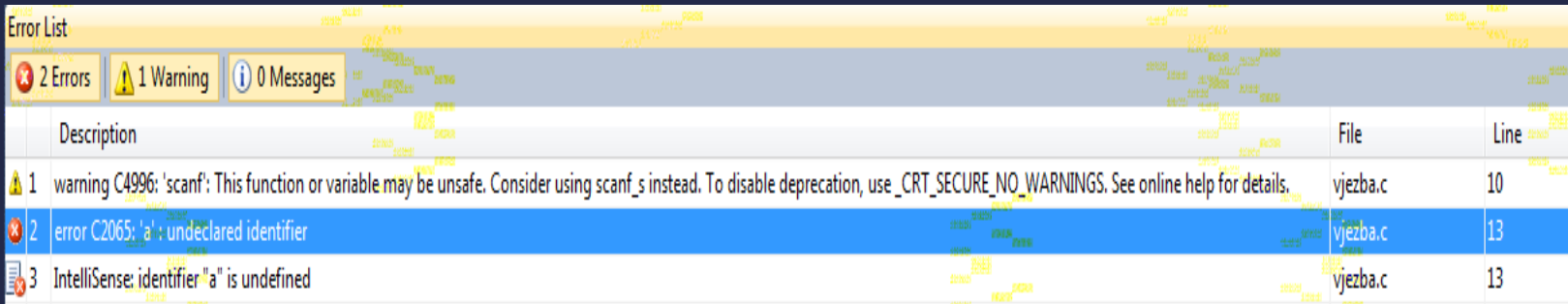
    printf("Vrijednost varijable iz bloka: x=%d", a);
}
}
```

Greška!



Primjer 1

- Pri pokušaju prevođenja javit će se pogreška.



	Description	File	Line
1	warning C4996: 'scanf': This function or variable may be unsafe. Consider using scanf_s instead. To disable deprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details.	vjezba.c	10
2	error C2065: 'a': undeclared identifier	vjezba.c	13
3	IntelliSense: identifier "a" is undefined	vjezba.c	13

- Varijabla deklarirana unutar bloka vidljiva je samo unutar tog bloka. U glavnoj funkciji varijabla **a** ne postoji. Zato se pri prevođenju javlja pogreška.

Globalne varijable

- Globalne varijable vrijede unutar glavne funkcije (unutar cijelog programa).
- Globalnoj varijabli je moguće pristupiti i unutar nekog bloka (blokovi su također dio glavne funkcije).

Primjer 2

- Primjer globalne varijable:
 - Deklarirati globalnu varijablu `x` u glavnoj funkciji i pridružiti joj vrijednost 2.
 - Potom unutar bloka učitati vrijednost 3 koju je potrebno pridružiti unutar globalne varijable `x`.
 - Nakon izlaska iz bloka ispisati vrijednost varijable `x`.

```
Vrijednost varijable x prije bloka: x=....  
Upisi cijeli broj:  
Vrijednost varijable x nakon bloka: x=....
```

Primjer 2

■ Rješenje

```
#include<stdio.h>

void main(void) {

    int x=2;
    printf("Vrijednost varijable x prije bloka: x=%d\n",x);

    {
        printf("Upisi cijeli broj:");
        scanf ("%d", &x);
    }

    printf("Vrijednost varijable x nakon bloka: x=%d\n",x);
}
```

C:\Windows\system32\cmd.exe

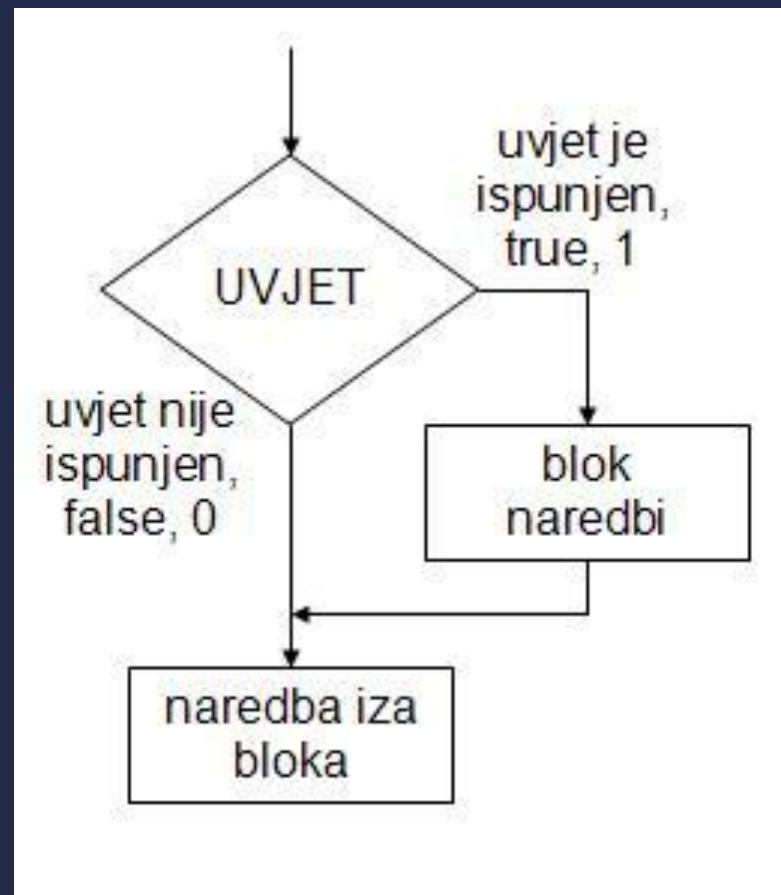
```
Vrijednost varijable x prije bloka: x=2
Upisi cijeli broj:3
Vrijednost varijable x nakon bloka: x=3
Press any key to continue . . .
```

Naredbe grananja

- Programska struktura grananja može se ostvariti naredbama:
 - *if*
 - *if – else*
 - *if – else if - else*
 - *switch – case.*

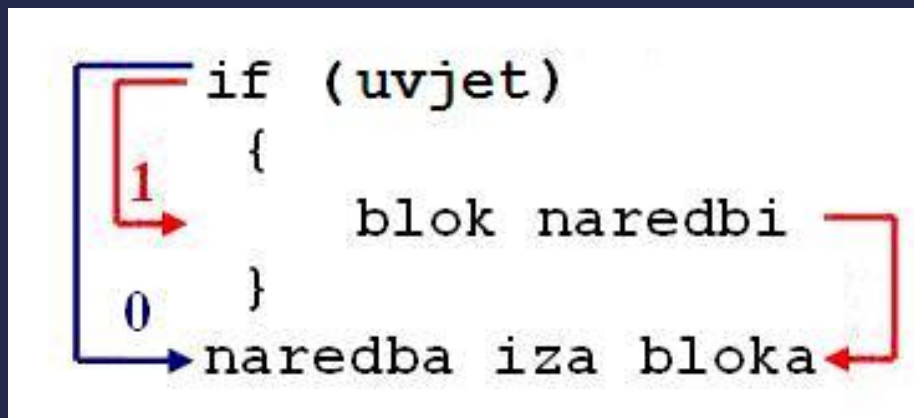
Jednostruko uvjetno grananje

- Jednostruko uvjetno grananje omogućava izvršenje bloka naredbi samo ako je zadani uvjet ispunjen.
- Ako uvjet nije ispunjen izvršava se prva naredba nakon bloka.



Jednostruko uvjetno grananje - if

- Za jednostruko uvjetno grananje rabi se naredba *if*.



- Uvjet je logički izraz, zapisuje se unutar para okruglih zagrada.
- Na kraju naredbe *if* ne stavlja se znak ;

Primjer 3

- Treba unijeti cijeli broj različit od 0 pa provjeriti da li je negativan ili pozitivan. U oba slučaja ispisati apsolutnu vrijednost broja.
- Ispis neka bude oblika:

```
Upisi cijeli broj razlicit od 0:  
Broj ... je ....  
Njegova apsolutna vrijednost je ....
```

Primjer 3

- Ako je ($a < 0$) izvršit će se prvi blok naredbi.
- Ako uvjet nije zadovoljen, prvi blok naredbi se preskače i izvođenje se programa nastavlja od prve naredbe iza bloka, a to je provjera drugog uvjeta ($a > 0$).
- Ako nije ispunjan niti drugi uvjet (za $a = 0$), drugi blok naredbi se preskače i program se nastavlja dalje izvoditi.

Primjer 3

■ Rješenje

```
#include<stdio.h>

void main(void){

    int a;
    printf("Upisi cijeli broj razlicit od 0:");
    scanf ("%d", &a);

    if (a<0)
    {
        printf("Broj %d je negativan.\nNjegova apsolutna vrijednost je %d", a, -a);
    }

    if (a>0)
    {
        printf("Broj %d je pozitivan.\nNjegova apsolutna vrijednost je %d\n", a, a);
    }

}
```

1. BLOK

2. BLOK

Primjer 3

- Za $(a > 0)$:

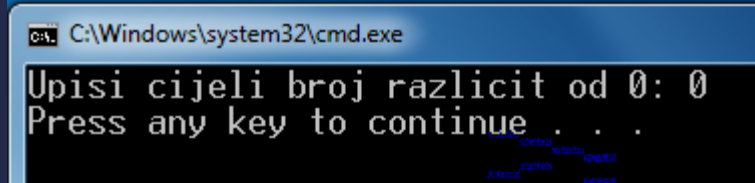
```
C:\Windows\system32\cmd.exe
Upisi cijeli broj razlicit od 0: 4
Broj 4 je pozitivan.
Njegova apsolutna vrijednost je 4
Press any key to continue . . .
```

- Za $(a < 0)$:

```
C:\Windows\system32\cmd.exe
Upisi cijeli broj razlicit od 0: -4
Broj -4 je negativan.
Njegova apsolutna vrijednost je 4
Press any key to continue . . .
```

Primjer 3

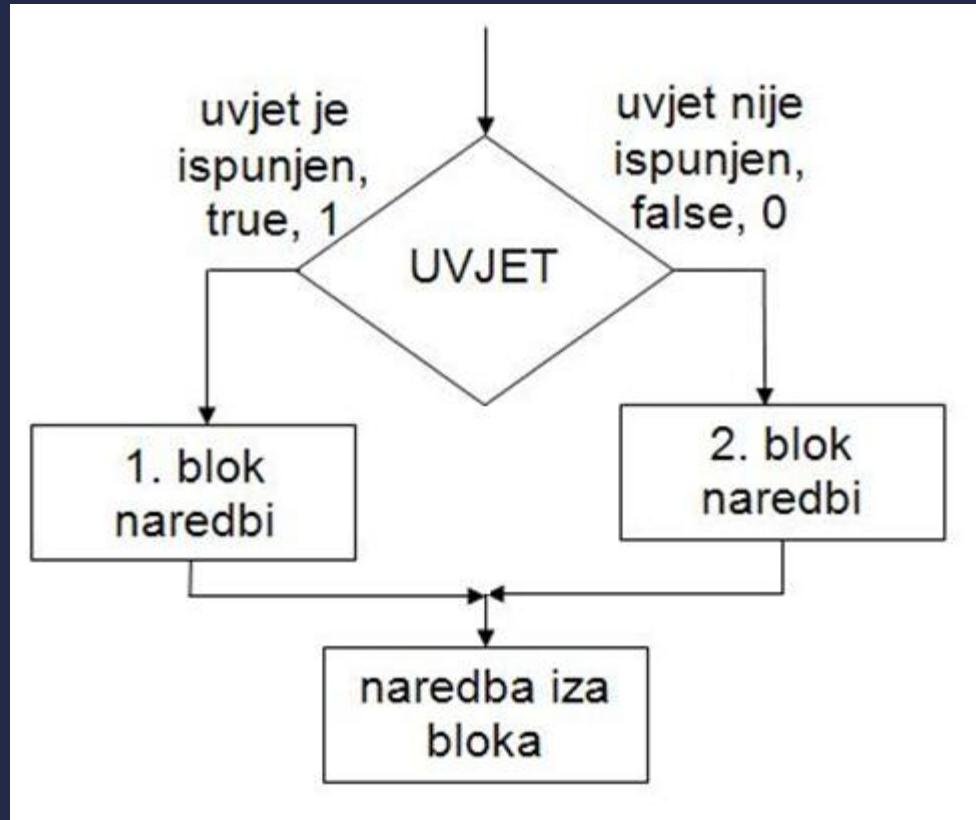
- Ako nije ispunjen niti prvi niti drugi uvjet (unesena je 0), budući da nije definirano ništa iza dva bloka program se prekida.



```
C:\Windows\system32\cmd.exe
Upisi cijeli broj razlicit od 0: 0
Press any key to continue . . .
```

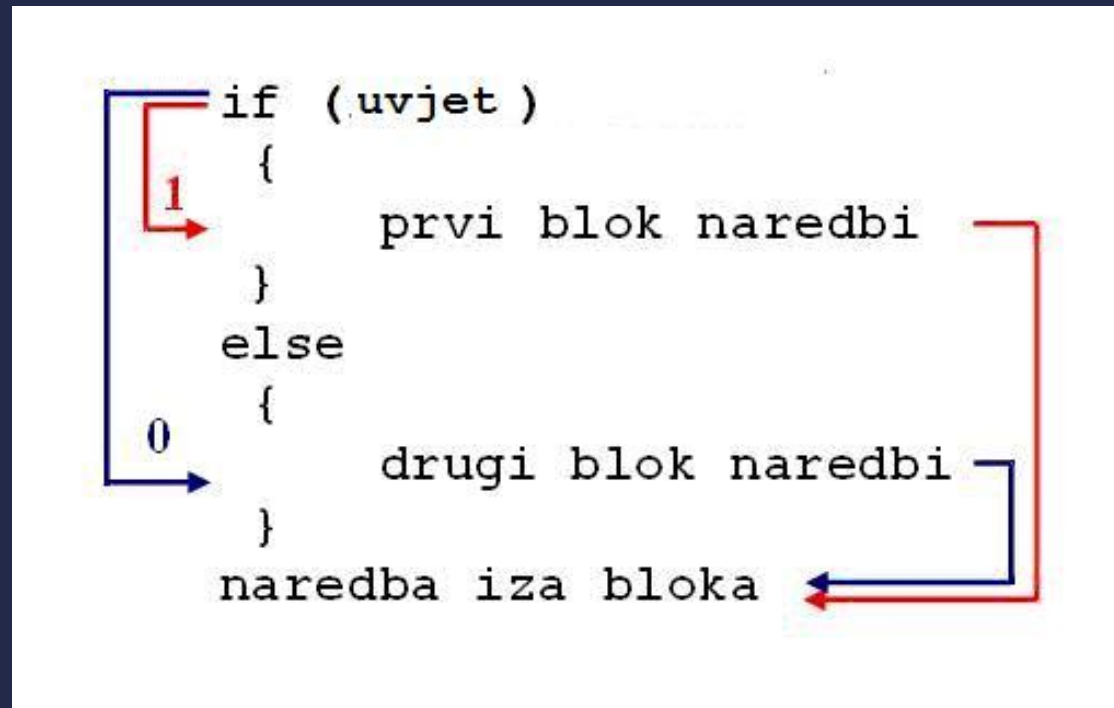
Dvostruko uvjetno grananje

- Dvostruko uvjetno grananje omogućava da se ovisno o ispunjenju postavljenog uvjeta izvodi jedan od dva neovisna bloka naredbi.



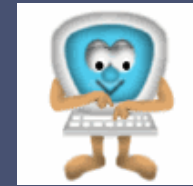
Dvostruko uvjetno grananje

- Za dvostruko uvjetno grananje rabi se naredba `if - else`.



Dvostruko uvjetno grananje

- Ako je vrijednost uvjeta logička istina izvodi se prvi blok. Nakon njegova završetka izvođenje se nastavlja od prve naredbe iza drugog bloka.
- Ako je vrijednost uvjeta logička neistina, preskače se prvi blok i izvodi se drugi blok (iza naredbe else). Nakon njegova završetka izvođenje se nastavlja od prve naredbe iza drugog bloka.



Primjer 4

- Primjer 3 treba riješiti uporabom dvostrukog uvjetnog grananja.
- Ispis neka bude oblika:

```
Upisi cijeli broj razlicit od 0:  
Broj ...je .... Njegova apsolutna  
vrijednost je ....
```

Primjer 4

- Primjer je riješen uz pomoć *if – else* naredbe.
- Ako je $(a < 0)$ izvršit će se prvi blok naredbi.
- Ako je $(a > 0)$, preskače se prvi blok i izvodi se drugi blok naredbi.

Primjer 4

■ Rješenje

```
#include<stdio.h>

void main(void) {

    int a;
    printf("Upisi cijeli broj razlicit od 0: ");
    scanf ("%d", &a);

    if (a<0)
    {
        printf("Broj %d je negativan.\nNjegova apsolutna vrijednost je %d", a, -a);
    }

    else
    {
        printf("Broj %d je pozitivan.\nNjegova apsolutna vrijednost je %d\n", a, a);
    }

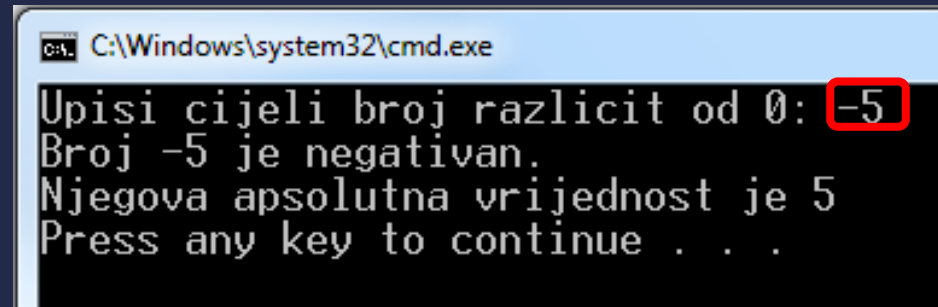
}
```

1. BLOK

2. BLOK

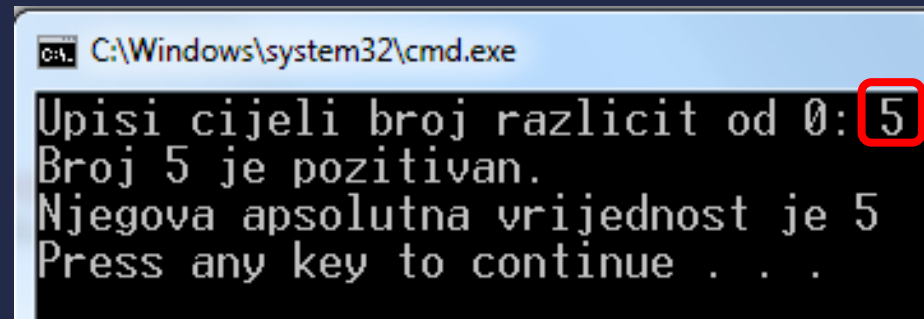
Primjer 4

- Za $(a < 0)$:



```
C:\Windows\system32\cmd.exe
Upisi cijeli broj razlicit od 0: -5
Broj -5 je negativan.
Njegova apsolutna vrijednost je 5
Press any key to continue . . .
```

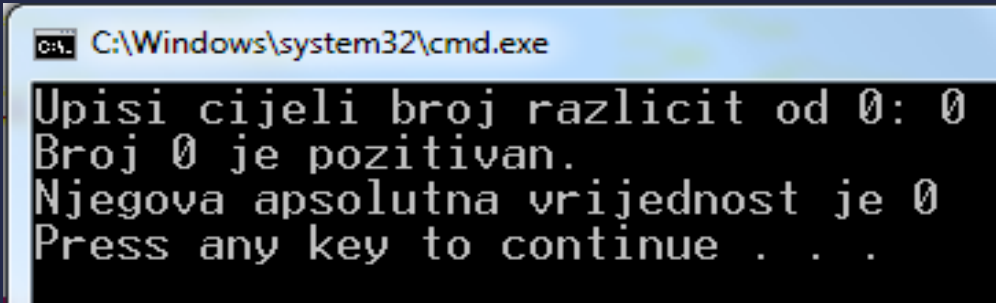
- Ako uvjet nije ispunjen:



```
C:\Windows\system32\cmd.exe
Upisi cijeli broj razlicit od 0: 5
Broj 5 je pozitivan.
Njegova apsolutna vrijednost je 5
Press any key to continue . . .
```

Pogreška unosa

- Što ako korisnik ne pročita uputu pažljivo, pa unese broj 0? Rezultat neće biti ispravan:



```
C:\Windows\system32\cmd.exe
Upisi cijeli broj razlicit od 0: 0
Broj 0 je pozitivan.
Njegova apsolutna vrijednost je 0
Press any key to continue . . .
```

- Je li moguće izbjeći ovu situaciju?

Pogreška unosa

- Uneseni broj treba provjeriti. Ako je unesena 0, korisnika valja upozoriti, a potom korisnik mora unijeti novi broj koji je različit od 0.
- Da bi to bilo moguće, potrebna je naredba koja omogućava nastavak odvijanja programa od odabrane naredbe.

Naredba goto

- Naredba **goto** omogućava **nastavak** odvijanja programa **od odabrane naredbe**.
- **Naredba na koju se želi skočiti**, od koje se želi nastaviti odvijanje programa, može biti **bilo gdje u programu**.
- Opći oblik naredbe:

```
goto oznaka_naredbe;
```

Naredba goto

- Naredbu od koje se želi nastaviti odvijanje programa treba označiti oznakom iza koje dolazi znak dvotočke.

```
oznaka_naredbe:naredba;
```

```
.....
```

```
.....
```

```
goto oznaka_naredbe;
```

Primjer 4a

■ Rješenje:

```
#include<stdio.h>

void main(void) {

    int a;
    upis:printf("Upisi cijeli broj razlicit od 0: ");
    scanf ("%d", &a);

    if(a==0) {
        printf("Pogresan unos. Ponovi!\n");
        goto upis;
    }

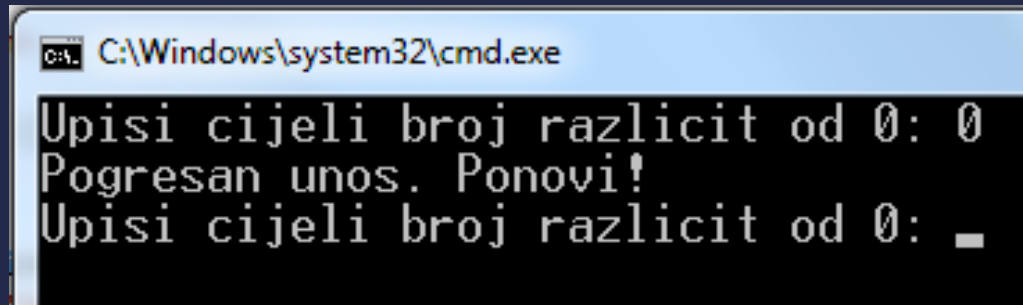
    if (a<0)
    {
        printf("Broj %d je negativan.\nNjegova apsolutna vrijednost je %d\n", a, -a);
    }

    else
    {
        printf("Broj %d je pozitivan.\nNjegova apsolutna vrijednost je %d\n", a, a);
    }

}
```

Primjer 4a

- Za $a==0$



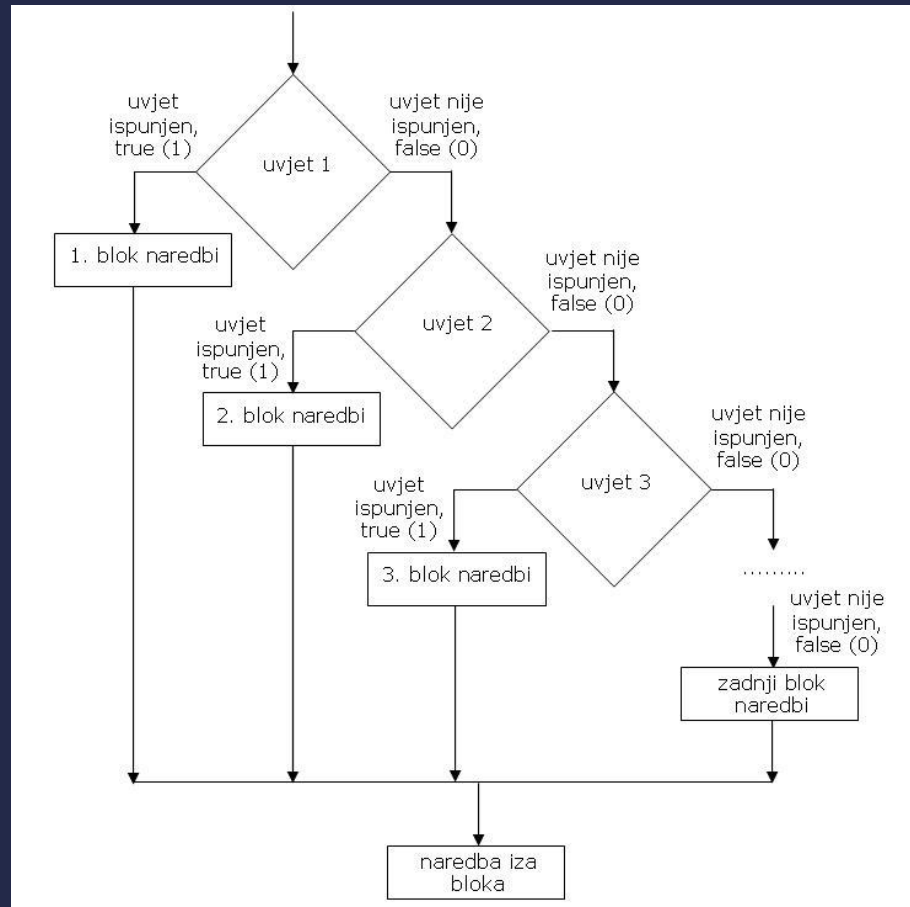
```
C:\Windows\system32\cmd.exe
Upisi cijeli broj razlicit od 0: 0
Pogresan unos. Ponovi!
Upisi cijeli broj razlicit od 0: _
```


goto

- Čestom uporabom naredbe goto teško je slijediti tijekom odvijanja programa što otežava otkrivanje pogrešaka.
- Naredbu goto stoga treba izbjegavati i nastojati zadatak riješiti na drugi način, pomoću petlji koje će biti objašnjene u nastavku.

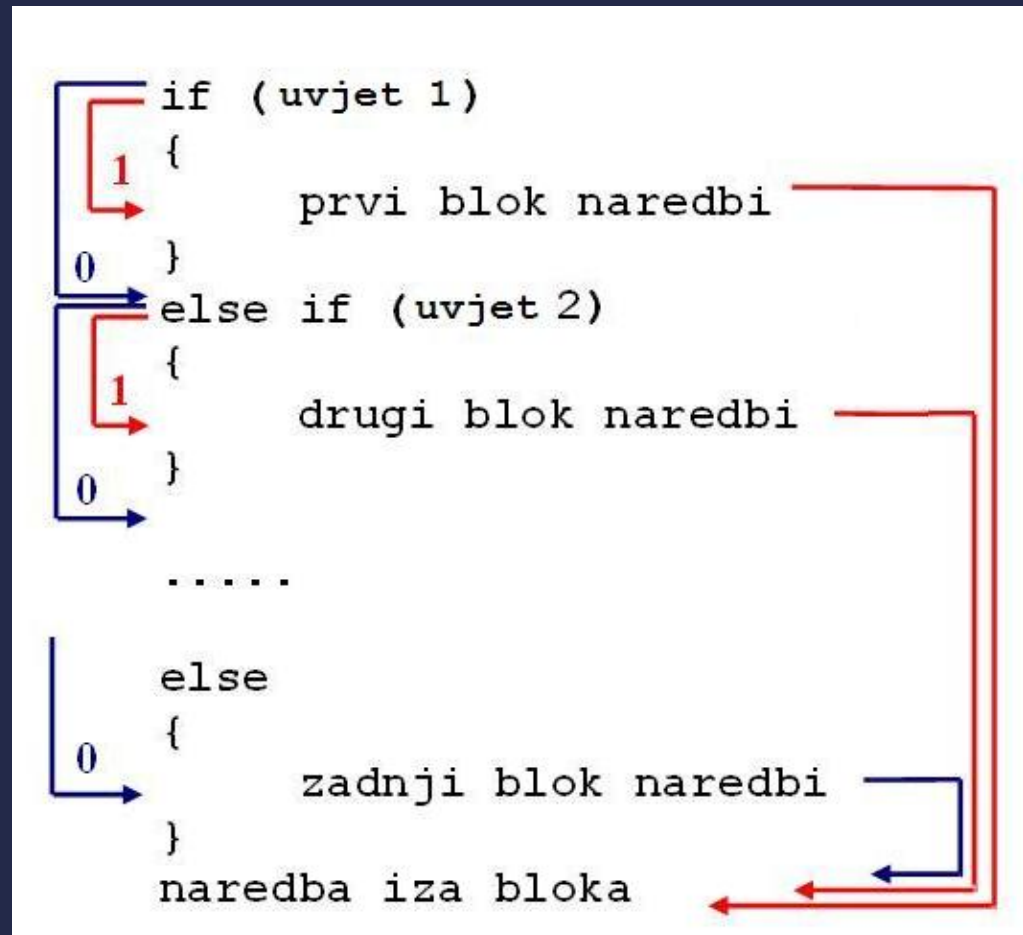
Višestruko uvjetno grananje

- Višestruko grananje omogućava ispitivanje više uvjeta.
- Ovisno o ispunjenju postavljenih uvjeta izvodi se odgovarajući blok naredbi.



Višestruko uvjetno grananje

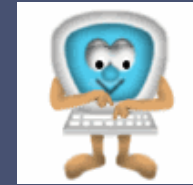
- Za višestruko uvjetno grananje rabi se naredba **if – else if - else**.
- Broj pustavljenih uvjeta nije ograničen.



Višestruko uvjetno grananje

- Ako je vrijednost prvog uvjeta logička istina, izvodi se prvi blok naredbi. Nakon njegova završetka izvođenje se nastavlja od prve naredbe iza zadnjeg bloka naredbi.
- Ako je vrijednost prvog uvjeta logička neistina, provjerava se drugi uvjet.
- Ako je on logička istina, izvodi se drugi blok naredbi, a potom prva naredba iza zadnjeg bloka naredbi.

Primjer 5



- Primjer 4 treba riješiti uporabom višestrukog uvjetnog grananja.
- Ispis neka bude oblika:

```
Upisi broj:  
Broj...je....Njegova apsolutna  
vrijednost je....  
ili  
Unio si 0. Apsolutna vrijednost od 0  
je 0.
```

Primjer 5

```
#include<stdio.h>

void main() {

    int a;

    printf("Upisi broj:");
    scanf("%d", &a);

    if (a>0) {
        printf("Broj %d je negativan.\n Njegova apsolutna vrijednost je %d.\n", a, a);
    }
    else if (a<0) {
        printf("Broj %d je negativan.\n Njegova apsolutna vrijednost je %d.\n", a, -a);
    }
    else {
        printf("Unio si 0. Apsolutna vrijednost od 0 je 0.\n");
    }
}
```

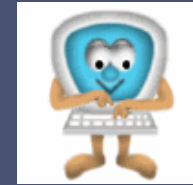
Primjer 5

- Ako se unese 0, poruka će biti:

```
0%  
Upisi broj: 0  
Unio si 0. Apsolutna vrijednost od 0 je 0.
```

- U ostalim slučajevima, provjera je ista kao u primjeru 4.

Primjer 6



- Treba upisati prirodni broj pa provjeriti da li je veći ili manji od 100, te da li je paran ili neparan.
- Ispis neka bude oblika:

```
Upisi prirodni broj:  
Uneseni broj je ... od 100  
Broj ... je .....
```


Primjer 6

- Višestrukim uvjetnim grananjem provjerava se da li je broj veći, manji ili jednak 100.
- Dvostrukim uvjetnim grananjem (neovisno o tome da li je broj veći ili manji od 100) provjerava se parnost broja.
- Parnost se provjerava operatorom modulo (ostatak dijeljenja s 2 se uspoređuje s 0).

Primjer 6

```
#include<stdio.h>

void main() {

    int a;
    upis:printf("Upisi prirodni broj:");
    scanf("%d", &a);

    if (a<=0) {
        printf("Pogrešan unos.Ponovi!\n");
        goto upis;
    }
    if (a<100) {
        printf("Broj %d je manji od 100.\n", a);
        if (a%2==0) {
            printf ("Broj %d je paran.\n", a);
        } else {
            printf ("Broj %d je neparan.\n", a);
        }
    }
    else if (a>100){
        printf("Broj %d je veći od 100.\n", a);
        if (a%2==0) {
            printf ("Broj %d je paran.\n", a);
        } else {
            printf ("Broj %d je neparan.\n", a);
        }
    }
    else {
        printf("Broj %d je jednak 100.\n", a);
    }
}
```

Primjer 6

■ Provjera:

```
C:\n
Upisi prirodni broj:435
Uneseni broj je veci od 100 i  neparan je.
Press any key to continue_

C:\n
Upisi prirodni broj:86
Uneseni broj je manji od 100 i  paran je.
Press any key to continue_

C:\n
Upisi prirodni broj:-9
Pogresan unos. Ponovi!
Upisi prirodni broj:_
```