

STRUKTURA GRANANJA



PROGRAMSKA STRUKTURA GRANANJA

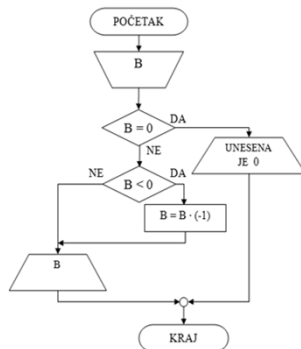
- Za rješavanje većine zadataka potrebne su programske strukture kod kojih **redoslijed izvršavanja naredbi ovisi o vrijednostima podataka koji se obrađuju**.
- Grananje je programska struktura koja omogućuje **različit tijek programa, ovisno o rezultatu postavljenog uvjeta**.

Sanda, 2014.

2

PRIMJER STRUKTURE GRANANJA

- Korisnik unosi cijeli broj, a zatim se računa apsolutna vrijednost tog broja i ispisuje rezultat.



Sanda, 2014.

3

BLOK NAREDBI

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

```
{
    int x;
    cout<<endl<<"Upisi cijeli broj u bloku:";
    cin>>x;
}
```

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

Sanda, 2014.

4

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**.

Sanda, 2014.

5

PRIMJER 23



- Treba deklarirati cjelobrojnu **varijablu x unutar zasebnog bloka**, pa joj pridružiti vrijednost. Pokušati ispisati vrijednost varijable x u glavnoj funkciji.

Upisi cijeli broj u bloku:
Vrijednost varijable iz bloka x=....

Sanda, 2014.

6

PRIMJER 23

```
#include<iostream>
using namespace std;
int main()
{
    {
        int x;
        cout<<endl<<"Upisi cijeli broj u bloku:";
        cin>>x;
    }
    cout<<endl<<"Vrijednost varijable iz bloka x="<<x<<endl;
    return 0;
}
```

BLOK

Sanda, 2014.

7

```
#include<iostream>
using namespace std;
int main()
{
    {
        int x;
        cout<<endl<<"Upisi cijeli broj u bloku:";
        cin>>x;
    }

    cout<<endl<<"Vrijednost varijable iz bloka x="
    <<x<<endl;
    return 0;
}
```

8

PRIMJER 23

- Pri pokušaju prevođenja javit će se **pogreška**.

```
Compiling...
x.cpp
x.cpp(10) : error C2065: 'x' : undeclared identifier
Error executing cl.exe.
x.exe - 1 error(s), 0 warning(s)
```



- Varijabla deklarirana unutar bloka **vidljiva je samo unutar tog bloka**. U glavnoj funkciji ona **ne postoji**.

Sanda, 2014.

9

NAREDBE GRANANJA

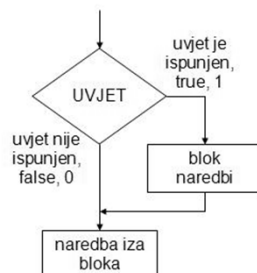
- Programska struktura grananja može se ostvariti naredbama:
 - if** (jednostruko uvjetno grananje)
 - if – else** (dvostruko uvjetno grananje)
 - if – else if – else** (višestruko uvjetno grananje)
 - switch – case**.

Sanda, 2014.

10

JEDNOSTRUKO UVJETNO GRANANJE

- Omogućava izvršenje bloka naredbi **samo ako je zadani uvjet ispunjen**.
- Ako **uvjet nije ispunjen** izvršava se **prva naredba nakon bloka**.



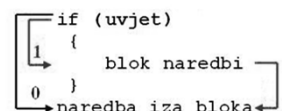
Sanda, 2014.

11

JEDNOSTRUKO UVJETNO GRANANJE - IF

- Za jednostruko uvjetno grananje rabi se **naredba if**.
- Uvjet je logički izraz**, zapisuje se **unutar para okruglih zagrada**.


```
if (uvjet)
{
    blok naredbi
}
```
- Na kraju naredbe **if** **ne stavlja se znak ;**



Sanda, 2014.

12

PRIMJER 24



- 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 različit od 0:
Broj ... je Njegova apsolutna vrijednost je

Sanda, 2014.

13

PRIMJER 24

- 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 izvođenje se nastavlja od naredbe **return 0**.

Sanda, 2014.

14

PRIMJER 24

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    cout<<"Upisi cijeli broj razlicit od 0: ";
    cin>>a;
    if(a<0)
    {
        cout<<"Broj "<<a<<" je negativan.\n";
        cout<<"Njegova apsolutna vrijednost je "<<-a<<endl;
    }
    if(a>0)
    {
        cout<<"Broj "<<a<<" je pozitivan.\n";
        cout<<"Njegova apsolutna vrijednost je "<<a<<endl;
    }
    return 0;
}
```

Sanda, 2014.

15

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    cout<<"Upisi cijeli broj razlicit od 0: ";cin>>a;
    if(a<0)
    {
        cout<<"Broj "<<a<<" je negativan.\n";
        cout<<"Njegova apsolutna vrijednost je "<<-a<<endl;
    }
    if(a>0)
    {
        cout<<"Broj "<<a<<" je pozitivan.\n";
        cout<<"Njegova apsolutna vrijednost je "<<a<<endl;
    }
    return 0;
}
```

16

PRIMJER 24

- Za ($a < 0$):

```
Upisi cijeli broj razlicit od 0: -456
Broj -456 je negativan. Njegova apsolutna vrijednost je 456
```

- Za ($a > 0$):

```
Upisi cijeli broj razlicit od 0: 456
Broj 456 je pozitivan. Njegova apsolutna vrijednost je 456
```

Sanda, 2014.

17

PRIMJER 24

- Ako nije ispunjen niti prvi niti drugi uvjet (unesena je 0), program se prekida.

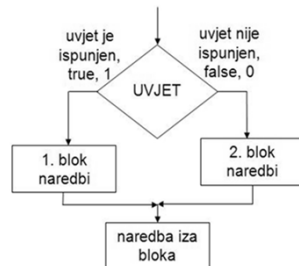
```
Upisi cijeli broj razlicit od 0: 0
Process returned 0 (0x0) execution time : 2.527 s
Press any key to continue.
```

Sanda, 2014.

18

DVOSTRUKO UVJETNO GRANANJE

- Omogućava da se ovisno o ispunjenju postavljenog uvjeta izvodi jedan od dva neovisna bloka naredbi.

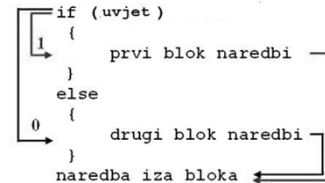


Sanda, 2014.

19

DVOSTRUKO UVJETNO GRANANJE

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



Sanda, 2014.

20

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.

Sanda, 2014.

21

PRIMJER 25

- Primjer 24 treba riješiti uporabom dvostrukog uvjetnog grananja.
- Ispis neka bude oblika:

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

Sanda, 2014.

22

PRIMJER 25

- 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.

Sanda, 2014.

23

PRIMJER 25

```

#include<iostream>
using namespace std;
int main()
{
    int a;
    cout<<"Upisi cijeli broj razlicit od 0: ";
    cin>>a;
    if(a<0)
    {
        cout<<"Broj "<<a<<" je negativan.\n";
        cout<<"Njegova apsolutna vrijednost je "<<-a<<endl;
    }
    else
    {
        cout<<"Broj "<<a<<" je pozitivan.\n";
        cout<<"Njegova apsolutna vrijednost je "<<a<<endl;
    }
    return 0;
}
  
```

Sanda, 2014.

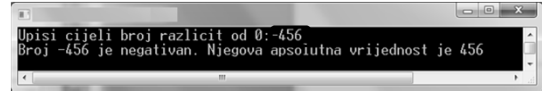
24

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    cout<<"Upisi cijeli broj razlicit od 0:"<<cin>>a;
    if(a<0)
    {
        cout<<"Broj "<<a<<" je negativan.\
Njegova apsolutna vrijednost je "<<-a<<endl;
    }
    else
    {
        cout<<"Broj "<<a<<" je pozitivan.\
Njegova apsolutna vrijednost je "<<a<<endl;
    }
    return 0;
}
```

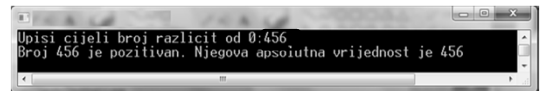
25

PRIMJER 25

- Za (a<0):



- Ako uvjet nije ispunjen:

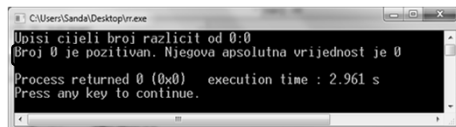


Sanda, 2014.

26

PRIMJER 25 - POGREŠKA UNOSA

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



- Bilo bi dobro izbjeći takvu situaciju provjerom unesenog broja. Ako je unesena 0, korisnika valja upozoriti, pa tražiti ponovni unos broja različitog od 0.

Sanda, 2014.

27

NAREDBA GOTO

- **goto** – omogućava nastavak odvijanja programa od odabrane naredbe.
- Naredba na koju se želi skočiti, tj. od koje se želi nastaviti odvijanje programa, može biti **bilo gdje u programu**, a treba biti označena **oznakom** iza koje dolazi **znak dvotočke**.

```
oznaka_naredbe:naredba;

.....

goto oznaka_naredbe;
```

Sanda, 2014.

28

PRIMJER 26

- U primjeru 25 dokinuti mogućnost da se broju 0 određuje predznak i apsolutna vrijednost.
- Ispis neka bude oblika:

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

Sanda, 2014.

29

PRIMJER 26

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    upis:cout<<"Upisi broj razlicit od 0:"<<cin>>a;
    if(a==0)
    {
        cout<<"Pogresan unos. Ponovi!"<<endl;
        goto upis;
    }
    if(a<0)
    {
        cout<<"Broj "<<a<<" je negativan.\
Njegova apsolutna vrijednost je "<<-a<<endl;
    }
    else
    {
        cout<<"Broj "<<a<<" je pozitivan.\
Njegova apsolutna vrijednost je "<<a<<endl;
    }
    return 0;
}
```

Sanda, 2014.

30

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    upis:cout<<"Upisi broj razlicit od 0:"<<cin>>a;
    if(a==0)
    {
        cout<<"Pogresan unos. Ponovi!"<<endl;
        goto upis;
    }
    if(a<0)
    {
        .
    }
}
```

31

PRIMJER 26

- Provjera programa unosom broja 0:

```
Upisi broj razlicit od 0:0
Pogresan unos. Ponovi!
Upisi broj razlicit od 0:
```

Sanda, 2014.

32

GOTO

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

Sanda, 2014.

33

IF - ELSE → KRATKI OBLIK

- Ako su **uvjet** i **naredbe uvjetnih blokova kratki**, umjesto **if - else** naredbi može se koristiti skraćeni oblik zapisa.

(uvjet) ? (1.blok naredbi) : (2.blok naredbi)

Koristi se kada uvjet i naredbe blokova stanu u jedan redak.

Sanda, 2014.

34

IF - ELSE → KRATKI OBLIK

- Ako se primijeni kratki oblik zapisa, primjer 25. mogao bi izgledati ovako:

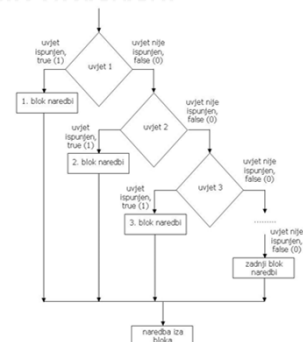
```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int a;
6     cout<<"Upisi cijeli broj razlicit od 0:"<<cin>>a;
7     (a<0) ? (cout<<-a) : (cout<<a);
8     return 0;
9 }
```

Sanda, 2014.

35

VIŠESTRUKO UVJETNO GRANANJE

- Omogućava **ispitivanje više uvjeta**.
- Ovisno o ispunjenju postavljenih uvjeta izvodi se **odgovarajući blok naredbi**.

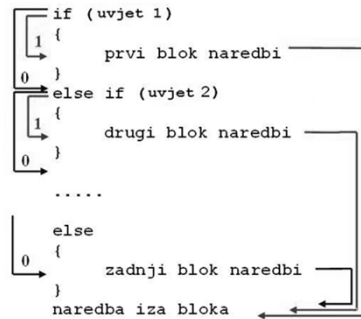


Sanda, 2014.

36

VIŠESTRUKO UVJETNO GRANANJE

- Broj postavljenih uvjeta **nije ograničen**.



Sanda, 2014.

37

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.

Sanda, 2014.

38

VIŠESTRUKO UVJETNO GRANANJE

- Ako je vrijednost **drugog uvjeta** logička **neistina**, provjerava se **treći uvjet**, i td.
- Provjere se tako redom nastavljaju sve do naredbe **else**.
- Ako do tada **niti jedan od uvjeta nije imao vrijednost logičke istine**, izvršit će se **zadnji blok** naredbi koji se nalazi iza naredbe **else**.

Sanda, 2014.

39

PRIMJER 27



- Primjer 24 treba riješiti uporabom višestrukog uvjetnog grananja.
- Ispis neka bude oblika:

Upisi cijeli broj razlicit od 0:
Broj ... je Njegova apsolutna vrijednost je
ili
Unio si broj 0.

Sanda, 2014.

40

PRIMJER 27

```

#include<iostream>
using namespace std;
int main()
{
    int a;
    cout<<"Upisi cijeli broj razlicit od 0:"<<cin>>a;
    if(a<0)
    {
        cout<<"Broj "<<a<<" je negativan.\
        Njegova apsolutna vrijednost je "<<-a<<endl;
    }
    else if (a>0)
    {
        cout<<"Broj "<<a<<" je pozitivan.\
        Njegova apsolutna vrijednost je "<<a<<endl;
    }
    else
    {
        cout<<"Unio si broj 0. "<<endl;
    }
    return 0;
}
    
```

Sanda, 2014.

41

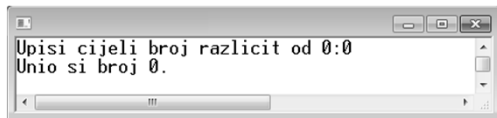
```

#include<iostream>
using namespace std;
int main()
{
    int a;
    cout<<"Upisi cijeli broj razlicit od 0:"<<cin>>a;
    if(a<0)
    {
        cout<<"Broj "<<a<<" je negativan.\
        Njegova apsolutna vrijednost je "<<-a<<endl;
    }
    else if (a>0)
    {
        cout<<"Broj "<<a<<" je pozitivan.\
        Njegova apsolutna vrijednost je "<<a<<endl;
    }
    else
    {
        cout<<"Unio si broj 0. "<<endl;
    }
    return 0;
}
    
```

42

PRIMJER 27

- Ako se unese 0, poruka će biti:



- U ostalim slučajevima, provjera je ista kao u primjeru 25.

Sanda, 2014.

43

PRIMJER 28



- Treba upisati prirodni broj pa provjeriti da li je veći ili manji od 100, te da li je paran ili neparan. Ako se unese broj manji ili jednak 0, ponoviti upis. Ispis neka bude oblika:

Upisi prirodan broj:
Uneseni broj ... je ... od 100 i
ili
Pogrešan unos. Ponovi!

Sanda, 2014.

44

PRIMJER 28

- 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).

Sanda, 2014.

45

- Provjerava da li je broj prirodan, a potom da li je manji, veći ili jednak 100.

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int a;
6     upis:cout<<"Upisi prirodni broj:"<<cin>>a;
7     if(a<=0)
8     {
9         cout<<"Pogresan unos. Ponovi!"<<endl;
10        goto upis;
11    }
12    if(a<100)
13    {
14        cout<<"Uneseni broj " <<a<<" je manji od 100 i ";
15    }
16    else if (a>100)
17    {
18        cout<<"Uneseni broj " <<a<<" je veci od 100 i ";
19    }
20    else
21    {
22        cout<<"Unio si broj 100. On je paran."<<endl;
23    }
24    return 0;
25 }
```

Sanda, 2014.

46

PRIMJER 28

```
if(a<100)
{
    cout<<"Uneseni broj " <<a<<" je manji od 100 i ";
    if (a%2==0)
        cout<<" paran je."<<endl;
    else
        cout<<" neparan je."<<endl;
}
else if (a>100)
{
    cout<<"Uneseni broj " <<a<<" je veci od 100 i ";
    if (a%2==0)
        cout<<" paran je."<<endl;
    else
        cout<<" neparan je."<<endl;
}
```

Sanda, 2014.

47

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    upis:cout<<"Upisi prirodni broj:"<<cin>>a;
    if(a<=0)
    {
        cout<<"Pogresan unos. Ponovi!"<<endl;
        goto upis;
    }
    if (a<100)
    {
        cout<<"Uneseni broj"<<a<<" je manji od 100 i ";
        if (a%2==0)
            cout<<" paran je."<<endl;
        else
            cout<<" neparan je."<<endl;
    }
}
```

48

```

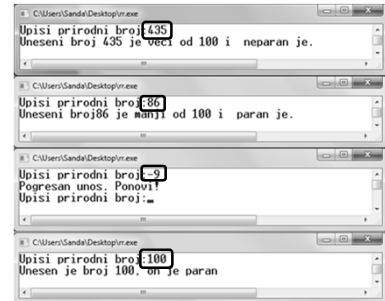
else if (a>100)
{
    cout<<"Uneseni broj "<<a<<" je veci od 100 i ";
    if (a%2==0)
        cout<<" paran je."<<endl;
    else
        cout<<" neparan je."<<endl;
}
else
{
    cout<<"Unesen je broj 100, on je paran"<<endl;
}
return 0;
}

```

49

PRIMJER 28

- Provjera programa:



Sanda, 2014.

50

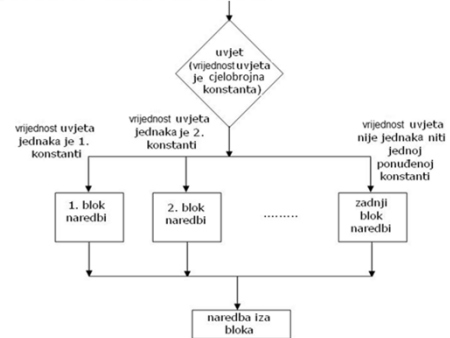
GRANANJE SWITCH-CASE

- Naredba **switch-case** omogućava **višestruko grananje** koje **ovisi o vrijednosti postavljenog uvjeta**.
- Uvjet** je **cjelobrojan izraz** ili **cjelobrojna varijabla** (izračun cjelobrojnog izraza ili vrijednost cjelobrojne varijable je cijeli broj - cjelobrojna konstanta).

Sanda, 2014.

51

GRANANJE SWITCH-CASE



Sanda, 2014.

52

GRANANJE SWITCH-CASE

- Vrijednost se uvjeta uspoređuje s nizom zadanih cjelobrojnih konstanti: konst1, konst2, konst3, itd.
- Ako je vrijednost uvjeta jednaka nekoj od zadanih konstanti, izvršava se blok naredbi pridružen toj konstanti.

```

switch (uvjet)
{
    case konstanta1:
        prvi blok naredaba
        break;
    case konstanta2:
        drugi blok naredaba
        break;
    case konstanta3:
        treci blok naredaba
        break;
    ....
    default:
        zadnji blok naredaba
}

```

Sanda, 2014.

53

GRANANJE SWITCH-CASE

- Po izvršenju bloka naredbi (kao primjer uzet je prvi blok), naredba **break** označava izlaz iz bloka **switch-case**.
- Ako vrijednost uvjeta nije jednaka niti jednoj od ponuđenih konstanti, izvršava se blok naredbi pridružen naredbi **default**.

- U slučaju izostavljanja naredbe **default** program će nastaviti izvršavanje prvom naredbom nakon **switch-case** bloka.

```

switch (uvjet)
{
    case konstanta1:
        prvi blok naredaba
        break;
    case konstanta2:
        drugi blok naredaba
        break;
    case konstanta3:
        treci blok naredaba
        break;
    ....
    default:
        zadnji blok naredaba
}

```

Sanda, 2014.

54

PRIMJER 29

- Treba izračunati ukupni otpor za otpore R1 i R2, ovisno o tome da li su spojeni serijski ili paralelno. Za odabir serijskog spoja korisnik upisuje 1, a za paralelu 2. Ako korisnik upiše broj koji nije 1 ili 2 ispisati upozorenje.

Otpor R1 (u omima):
Otpor R2 (u omima):
Za serijski spoj otpora upisi 1, a za paralelni 2:
Ako se otpori od ... oma i ... oma spoje u ... ukupni je otpor ... oma.
ili
Pogrešan unos! Unesi 1 ili 2!

Sanda, 2014.

55

PRIMJER 29

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     float R, R1, R2;
6     int unos;
7     cout<<"Otpor R1 (u omima):"; cin>>R1;
8     cout<<"Otpor R2 (u omima):"; cin>>R2;
9     cout<<"Za spoj otpora u seriju upisi 1, a za paralelu 2:"; cin>>unos;
10    switch (unos)
11    {
12        case 1:
13            R=R1+R2;
14            cout<<"Ako se otpori od "<<R1<<" oma i "<<R2
15            <<" oma spoje u seriju ukupni je otpor "<<R<<" oma."<<endl;
16            break;
17        case 2:
18            R=(R1*R2)/(R1+R2);
19            cout<<"Ako se otpori od "<<R1<<" oma i "<<R2
20            <<" oma spoje u paralelu ukupni je otpor "<<R<<" oma."<<endl;
21            break;
22        default:
23            cout<<"Pogresan unos. Unesi 1 ili 2"<<endl;
24    }
25    return 0;
26 }
```

Sanda, 2014.

6

```
#include<iostream>
using namespace std;
int main()
{
    float R, R1, R2;
    int unos;
    cout<<"Otpor R1 (u omima):";cin>>R1;
    cout<<"Otpor R2 (u omima):";cin>>R2;
    cout<<"Za spoj otpora u seriju upisi 1, a za paralelu 2:";
    cin>>unos;
    switch (unos)
    {
        case 1:
            R=R1+R2;
            cout<<"Ako se otpori od "<<R1<<" oma i "<<R2
            <<" oma spoje u seriju ukupni je otpor "
            <<R<<" oma."<<endl;
            break;
    }
```

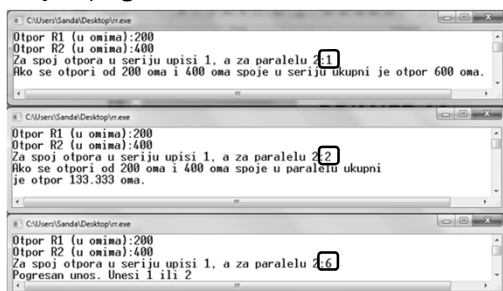
57

```
case 2:
    R= (R1*R2) / (R1+R2);
    cout<<"Ako se otpori od "<<R1<<" oma i "
    <<R2 <<" oma spoje u paralelu ukupni \
    je otpor "<<R<<" oma."<<endl;
    break;
default:
    cout<<"Pogresan unos. Unesi 1 ili 2";
}
return 0;
}
```

58

PRIMJER 29

- Provjera programa:



Sanda, 2014.

59