Početna grupa – osnovna škola, 03.02.2018.

Zadatak 1

Napisati program koji učitava cio broj n i štampa sljedeću „piramidu” brojeva (dat je primjer za n =5).

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

```
#include <iostream>

#include <iostream>

using namespace std;
int main()
{
    int n;
    int red; // oznacava posljednji broj koji stampamo u trenutnom redu
    int cnt; // trenutni broj koji stampamo u redu
    cout << "Unesite prirodan broj:";
    cin >> n;
    red = n;
    while(red >= 1)
    {
        cnt = 1;
        while(cnt <= red)
        {
            cout << cnt << " ";
            cnt++;
        }
        red--; // red -= 1; red = red - 1;
        cout << endl; // odstampani svi brojevi reda - predji u novi red
    }
    return 0;
}
```

Zadatak 2

Napisati program koji učitava cio broj n i štampa sljedeću „piramidu” brojeva (dat je primjer za n =5).

```
5
5 4
5 4 3
5 4 3 2
5 4 3 2 1
5 4 3 2
5 4 3
5 4
5
```

```
#include <iostream>

using namespace std;
int main()
{
    int n;
    int red; // oznacava posljednji broj koji stampamo u trenutnom redu
    int cnt; // trenutni broj koji stampamo u redu
    cout << "Unesite prirodan broj:";
    cin >> n;
    red = n;
    while(red >= 1)
    {
        cnt = n;
        while(cnt >= red)
        {
            cout << cnt << " ";
            cnt--;
        }
        red--; // red -= 1; red = red - 1;
        cout << endl; // odstampani svi brojevi reda - predji u novi red
    }

    // zavrseno stampanje prvih n redova, prelazimo na poslednjih n-1 redova

    red = 2;
    while(red <= n)
    {
        cnt = n;
        while(cnt >= red)
        {
            cout << cnt << " ";
            cnt--;
        }
        red++; // red -= 1; red = red - 1;
        cout << endl; // odstampani brojevi u redu - predji u novi red
    }
    return 0;
}
```

Domaći zadatak

1. Cio broj ***x*** je izveden iz broja ***y*** ako se broj |***x***| može dobiti brisanjem tačno jedne cifre iz broja ***y***. Napisati program koji prvo učitava jedan cio broj ***x***, a zatim učitava cijele brojeve sve dok se ne učita broj koji je po apslutnoj vrijednosti veći od 2000000000 i za svaki od učitanih brojeva ***y*** štampa poruku „GOOD" ako je ***y*** izveden iz ***x*** ili „BAD" ako se ***y*** ne može izvesti iz ***x***.

2. Koristeći samo dva karaktera '.' (tačku) i '*' (zvjezdicu) štampati "šahovsku tablu". Prvi simbol treba da bude *. Program učitava dva pozitivna cijela broja m i n koji predstavljaju broj redova i broj kolona šahovske table. Vidi tabelu sa primjerima.

	Primjer 1	Primjer 2	Primjer 3
Ulaz	`3 1`	`4 4`	`2 5`

Izlaz	`*`
	`.`
	`*`
		`*.*.`
		`.*.*`
		`*.*.`
		`.*.*`
			`*.*.*`
			`.*.*.`

3. Koristeći samo dva karaktera '.' (tačku) i '*' (zvjezdicu) štampati oblik "okvira za sliku". Program učitava dva pozitivna cijela broja m i n koji predstavljaju broj redova i broj kolona treba da ima okvir. Vidi tabelu sa primjerima.

	Primjer 1	Primjer 2	Primjer 3
Ulaz	`3 1`	`4 4`	`2 5`

Izlaz	`*`
	`*`
	`*`
		`****`
		`*..*`
		`*..*`
		`****`
			`*****`
			`*****`

4. (*) Koristeći samo dva karaktera '.' (tačku) i '*' (zvjezdicu) štampati oblik "mreže" (engl. grid-like pattern). Program učitava dva pozitivna cijela broja m i n koji predstavljaju broj redova i broj kolona mreže. Svako polje mreže ima 4 unutrašnja polja (tj. sadrži po 4 tačke). Vidi tabelu sa primjerima.

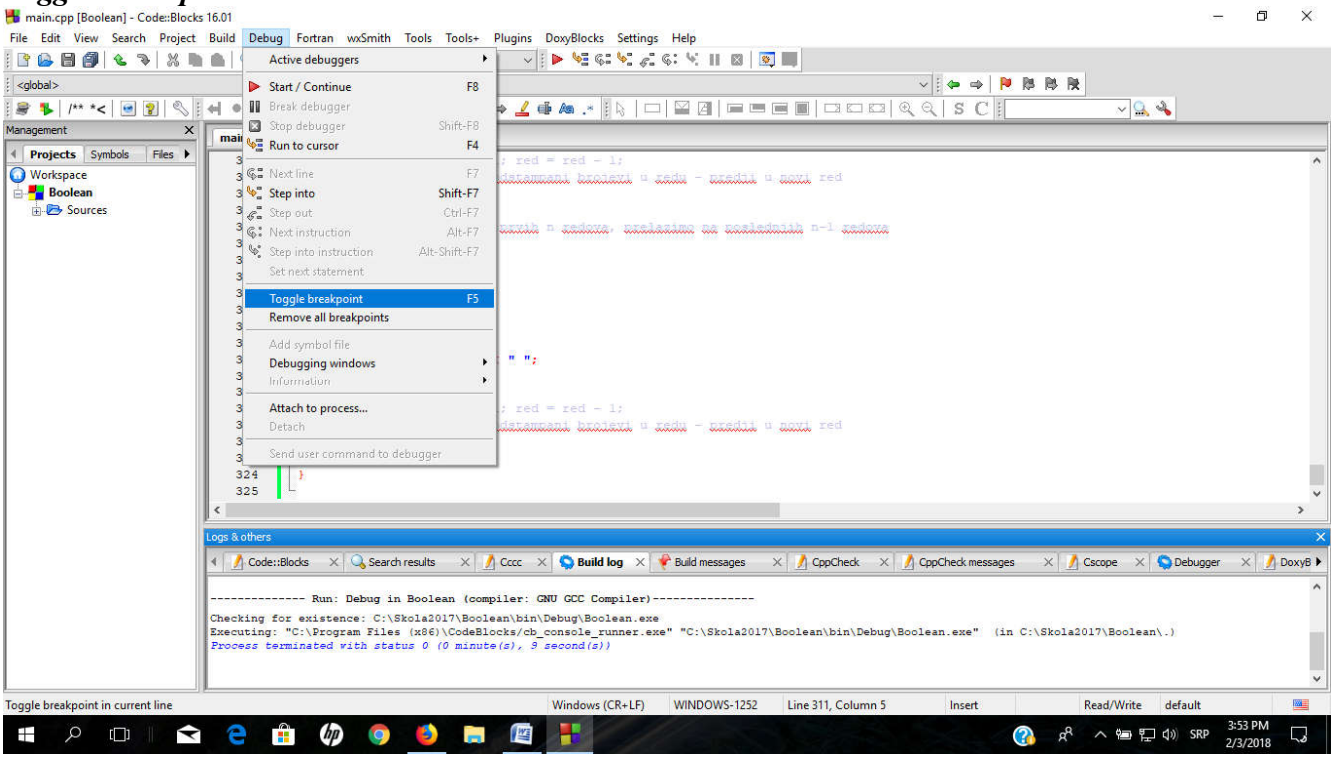
	Primjer 1	Primjer 2	Primjer 3
Ulaz	`3 1`	`4 4`	`2 5`

Izlaz	`****`
	`*..*`
	`*..*`
	`****`
	`*..*`
	`*..*`
	`****`
	`*..*`
	`*..*`
	`****`
		`**********`
		`*..*..*..*..*`
		`*..*..*..*..*`
		`**********`
		`*..*..*..*..*`
		`*..*..*..*..*`
		`**********`
		`*..*..*..*..*`
		`*..*..*..*..*`
		`**********`
			`**********`
			`*..*..*..*..*..*`
			`*..*..*..*..*..*`
			`**********`
			`*..*..*..*..*..*`
			`*..*..*..*..*..*`
			`**********`

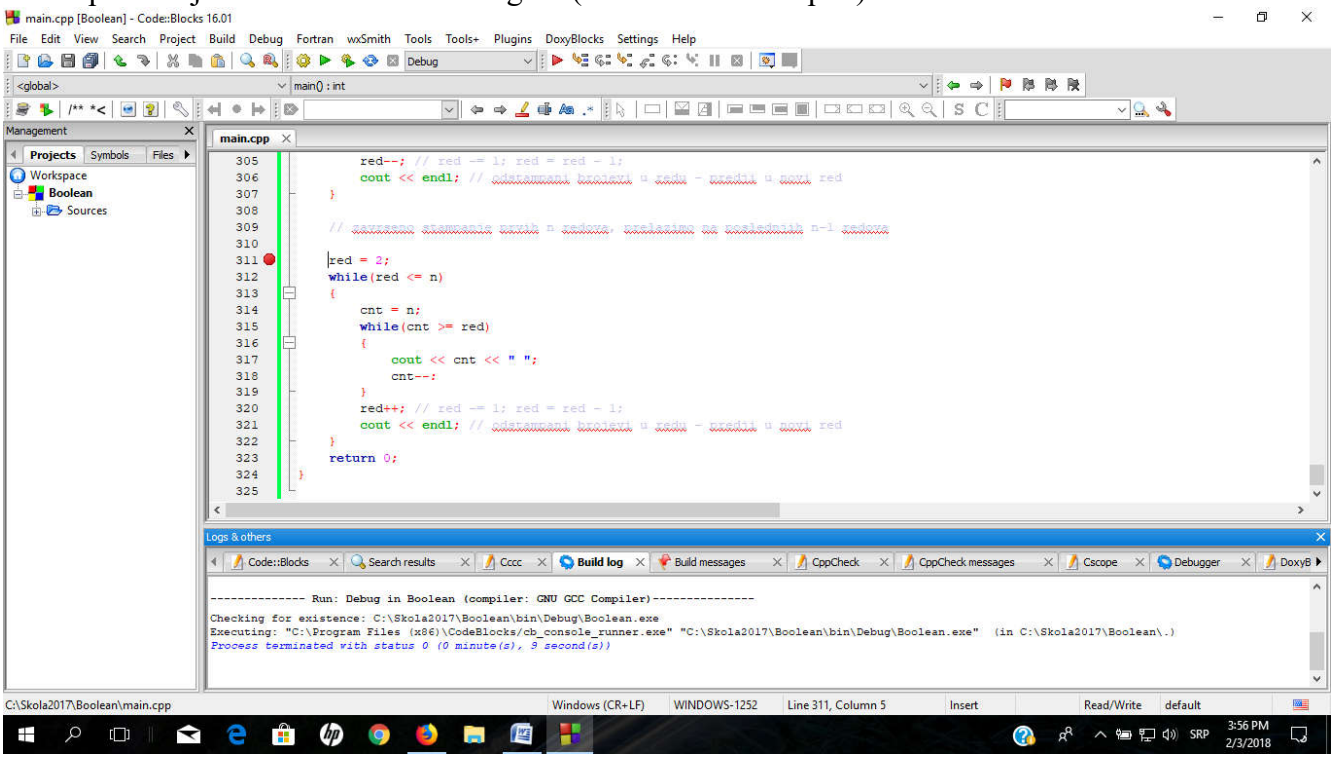
Pronalaženje grešaka u programu (debugging)

Prva greška u programu: (http://www.computerhistory.org/tdih/September/9/)

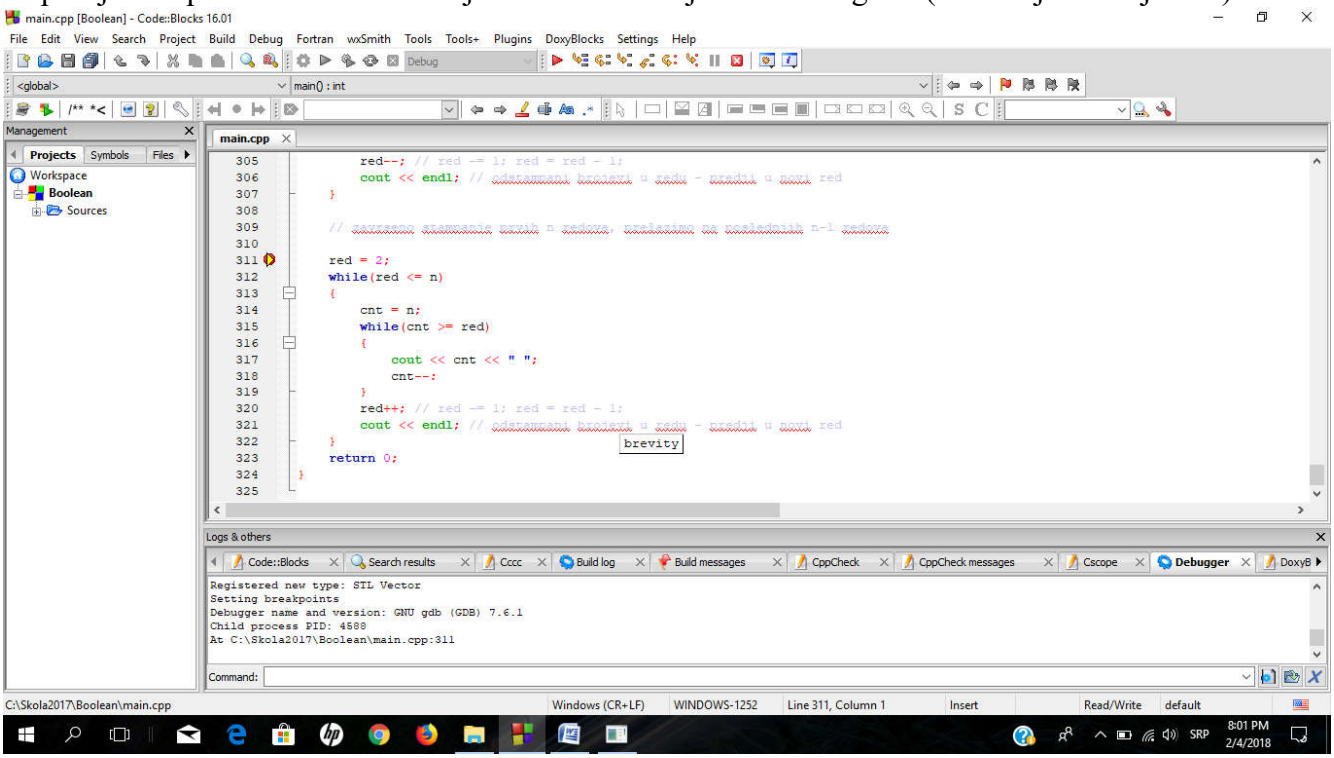
Postavljanje tačke prekida: klik na red gdje želite da bude tačka prekida, a pa zatim ***F5*** ili ***Debug-Toggle Breakpoint***.



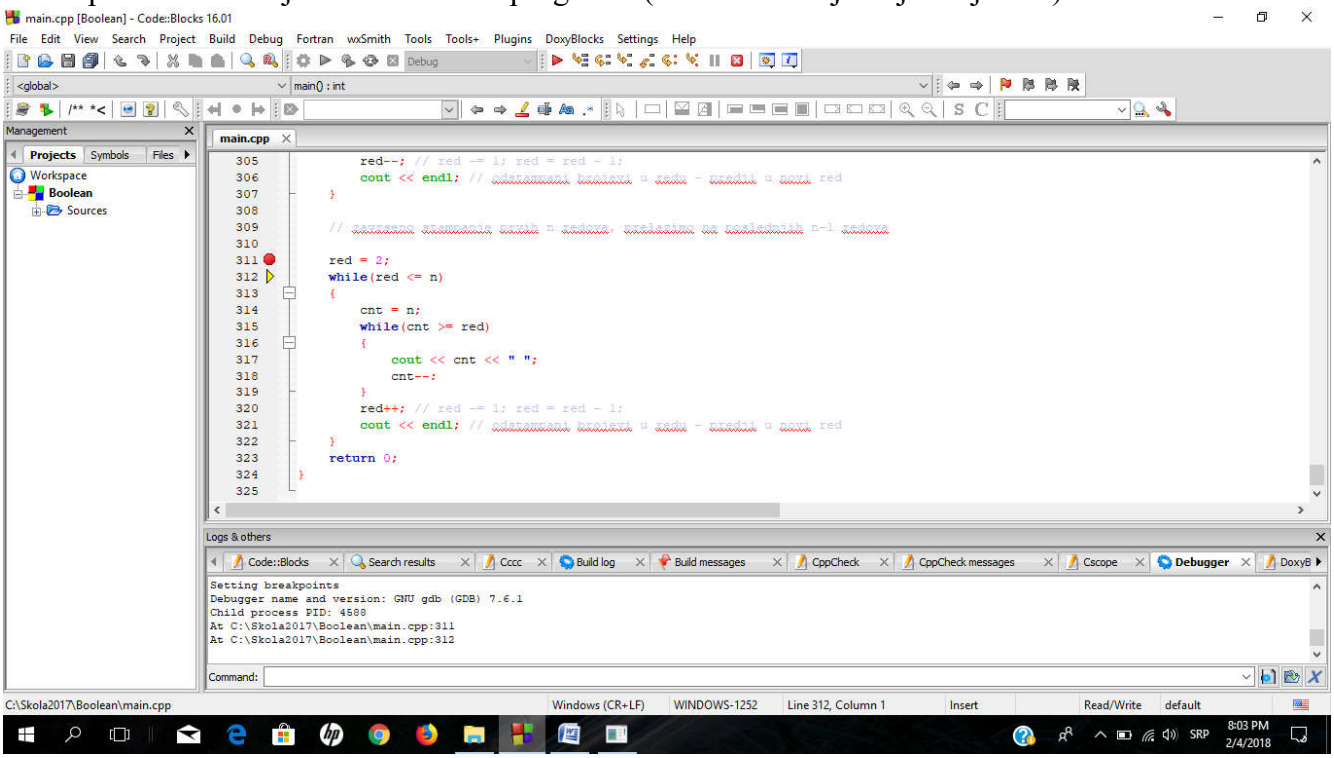
Tačka prekida je označena crvenim krugom (red 311 na slici ispod):



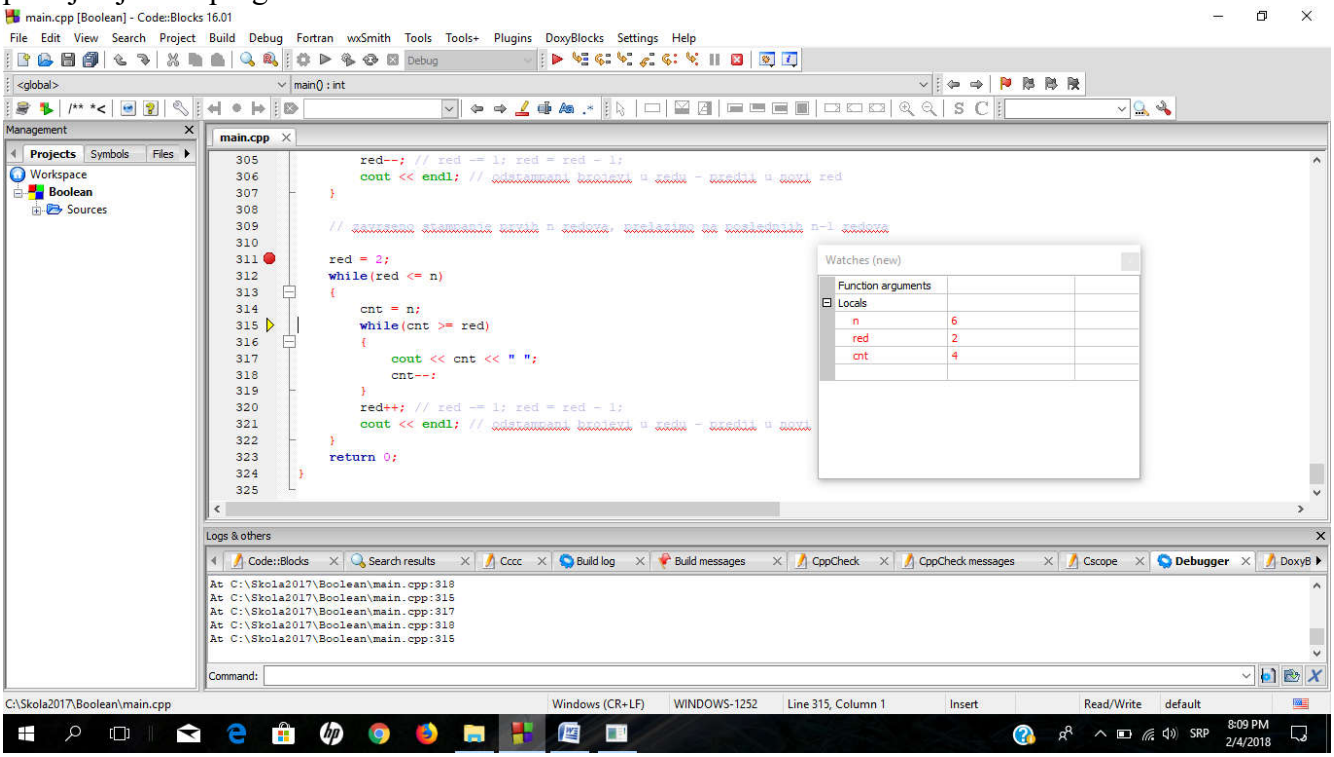
Uklanjanje tačke prekida: klik na red koji sadrži tačku prekida, pa zatim ***F5*** ili ***Debug-Toggle Breakpoint***.
Pokretanje programa: stavka menija ***Debug – Start/Continue*** ili ***F8***. Izvršavanje programa se zaustavlja na prvoj tački prekida. Trenutna linija kofa označena je žutim trouglom (na slici je to linija 311) .



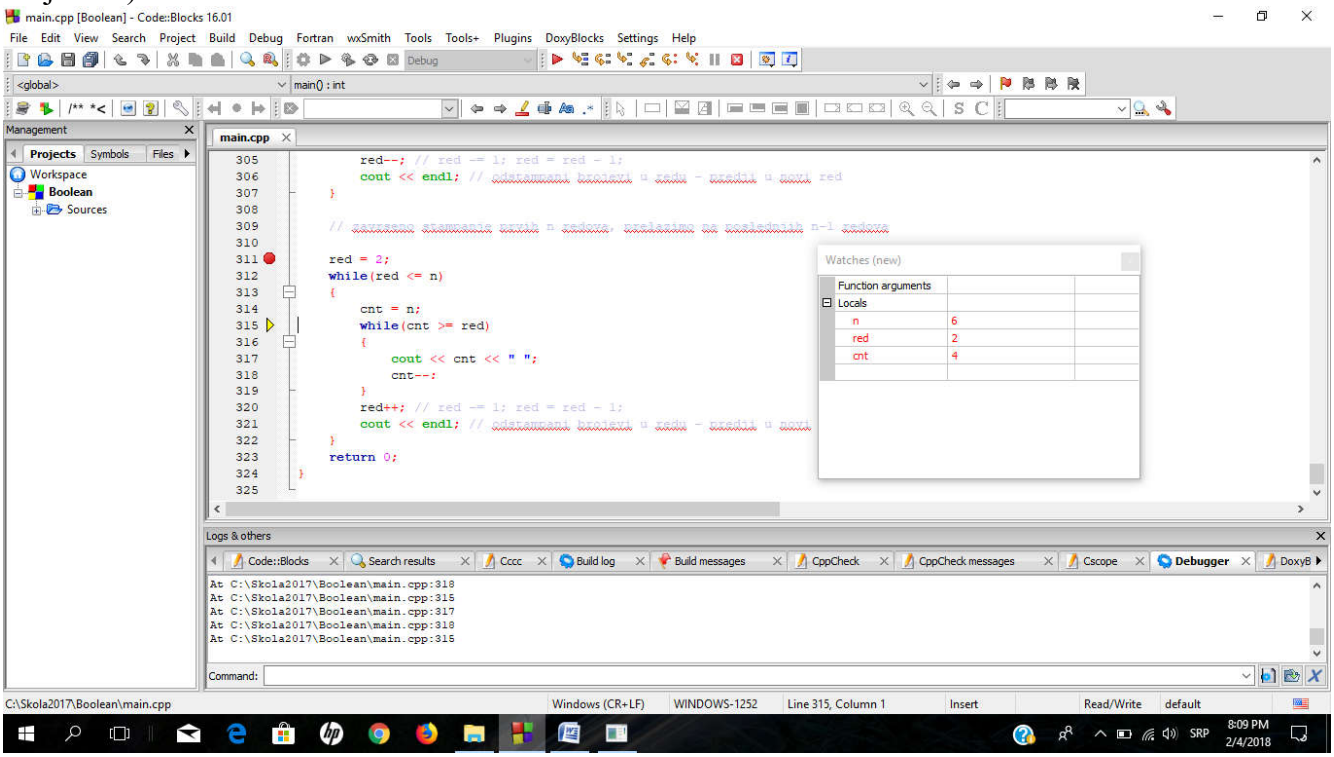
Sa ***F7*** prelazimo na sljedeću naredbu u programu (u našem slučaju to je linija 312).



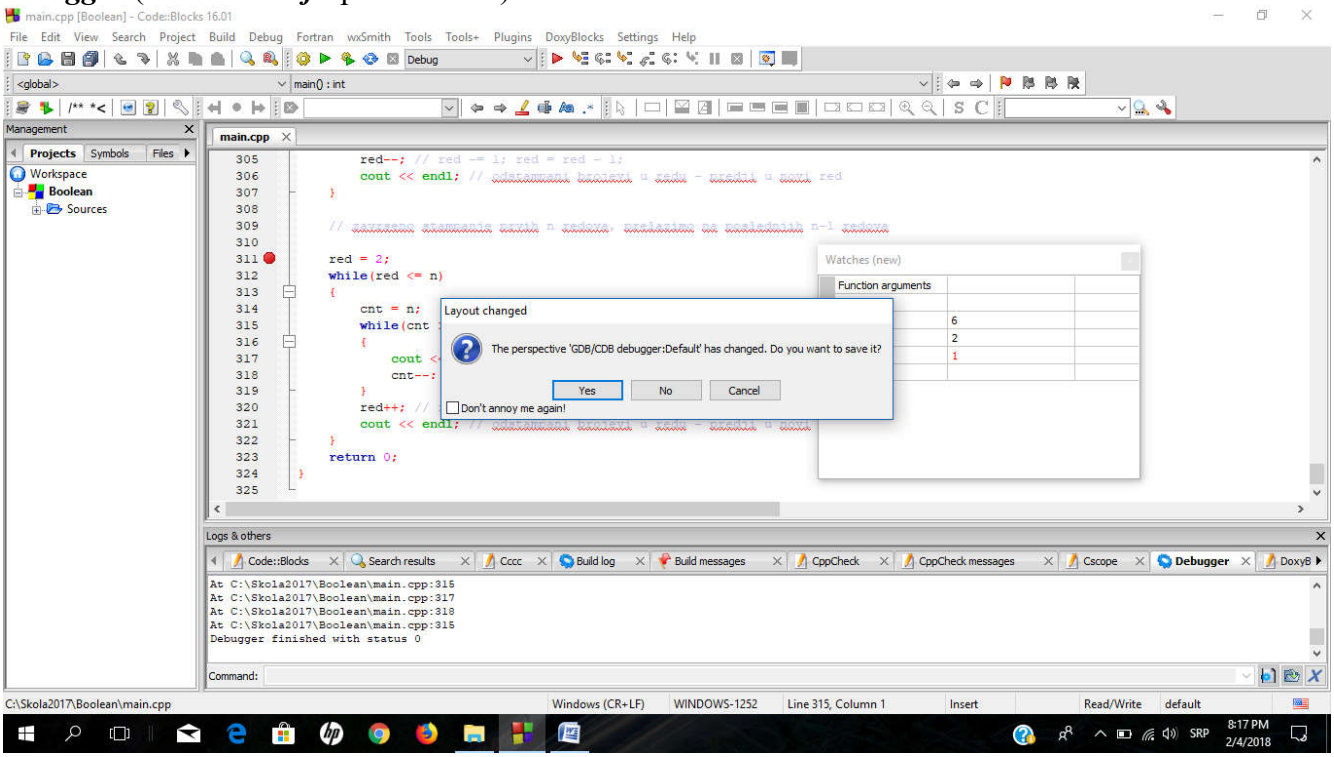
Da bi vidjeli promjenljive u programu i njihove trenutne vrijednosti, izaberite u meniju ***Debug-Debugging windows- Watches***. Otvara se novi prozor ***Watchec*** koji pokazuje trenutne vrijednosti promjenljivih u programu.



Poslije nekoliko koraka, kada dođemo do kraja ciklusa `while (cnt>=red)` (tj. kada `cnt` ne bude veće od `red`), ***F7*** vas vraća na početak ciklusa (na slici ispod, ***F7*** je napravio skok sa linije 318 na liniju 315).



Kada završite sa debug-ovanjem vašeg programa, prekinite prosec klikom na meni ***Debug-Stop debugger*** (ili držite ***Shifti*** pritisnite ***F8***):



U dijalog-prozoru ***Layout changed***, kliknite na *Yes* da bi *CodeBlocks* sačuvao raspored prozora za sljedeći proces debug-ovanja.