

Programming, Data Structures and Algorithms
Prof. Shankar Balachandran
Department of Computer Science and Engineering
Indian Institute Technology, Madras

Module – 12
Lecture - 12
Module Debugging: Demo Module
Contents
Demo of debugging

(Refer Slide Time: 00:21)

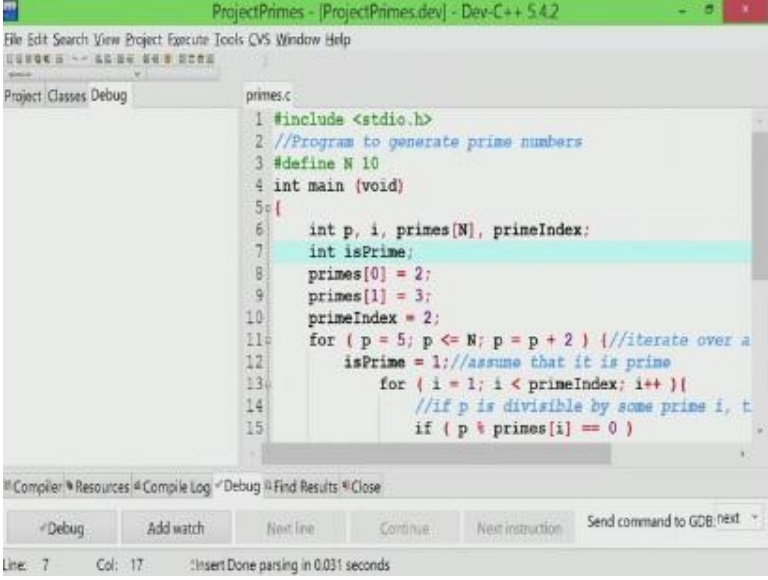
Problem 3.1: Finding All Prime Numbers $\leq N$

□ Observations:

- 2 is the only even prime number
- 3 is the smallest odd prime number
- To find out if a number p is prime or not, it is enough to check if p is divisible by all primes less than p
 - If some prime number $i < p$ divides p , then p is composite
 - If there is no i such that i divides p fully, then p is prime

Hello all. In this small demo of the debugging environment, I want to show you some specific features in debugging and how things work with debuggers. So, we will go back to this program, where we were looking at prime numbers. Just to recollect, we declare the 2 and 3 will be prime numbers. And for the other prime numbers, we will actually go and evaluate based on this small logic that we discussed. If some... For a number p , we are going to check against all the prime numbers that are less than it. And if it is even divisible by one of them, we will declare that, that it is composite; otherwise at the end of checking all the prime numbers, if the number is still not divisible, then we will say that, p is prime. So, let us switch to the programming environment.

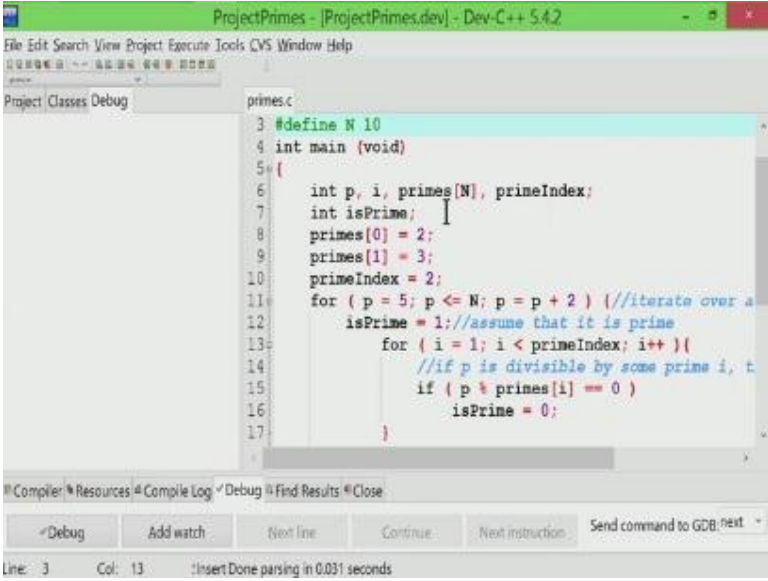
(Refer Slide Time: 00:59)



```
1 #include <stdio.h>
2 //Program to generate prime numbers
3 #define N 10
4 int main (void)
5 {
6     int p, i, primes[N], primeIndex;
7     int isPrime;
8     primes[0] = 2;
9     primes[1] = 3;
10    primeIndex = 2;
11    for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12        isPrime = 1; //assume that it is prime
13        for ( i = 1; i < primeIndex; i++ ){
14            //if p is divisible by some prime i, t
15            if ( p % primes[i] == 0 )
```

We saw this program earlier.

(Refer Slide Time: 01:04)

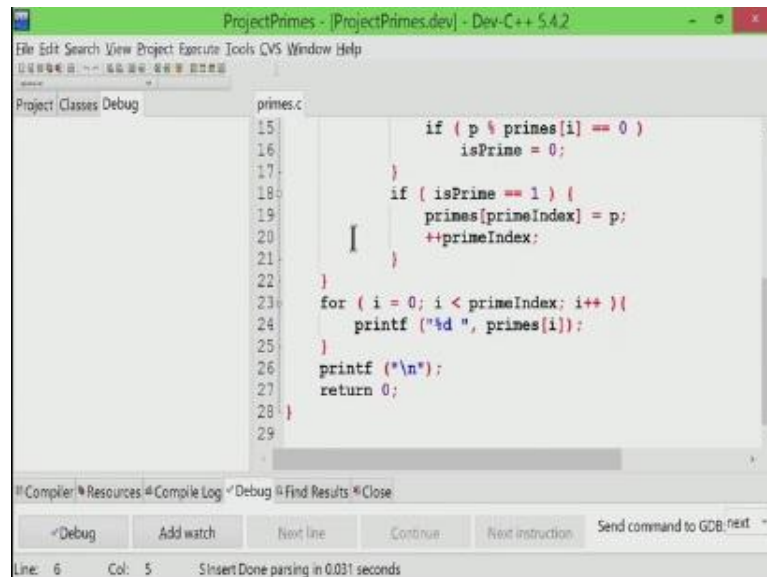


```
3 #define N 10
4 int main (void)
5 {
6     int p, i, primes[N], primeIndex;
7     int isPrime;
8     primes[0] = 2;
9     primes[1] = 3;
10    primeIndex = 2;
11    for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12        isPrime = 1; //assume that it is prime
13        for ( i = 1; i < primeIndex; i++ ){
14            //if p is divisible by some prime i, t
15            if ( p % primes[i] == 0 )
16                isPrime = 0;
17        }
```

And the only modification that we have is I have defined N to be 10, so that we can quickly see what is happening. So, one aspect of this debugging is essentially that, if there is a small error in the program, how do we find it out? So, I can always go and run the program multiple times or I can put printf statements in multiple locations and track. But, that is not the best thing to do. So, if the programs get very large, then the printf's can be very annoying. So, instead, many ID is including del C plus plus; give a very nice

mechanism for debugging. And that is what we are going to do. So, there are two things that we are going to do when we have a debugger. The first thing that we are going to do is set of border called break points. So, generally, what happens is your program starts executing at line number 6 let us say.

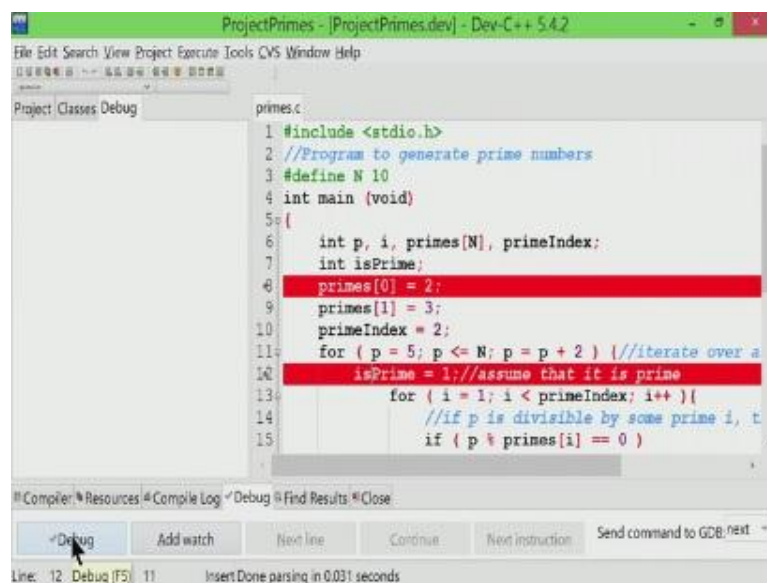
(Refer Slide Time: 01:51)



```
ProjectPrimes - [ProjectPrimes.dev] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project: Classes Debug
primes.c
15:         if ( p % primes[i] == 0 )
16:             isPrime = 0;
17:         }
18:         if ( isPrime == 1 ) {
19:             primes[primeIndex] = p;
20:             ++primeIndex;
21:         }
22:     }
23:     for ( i = 0; i < primeIndex; i++ ){
24:         printf ("%d ", primes[i]);
25:     }
26:     printf ("\n");
27:     return 0;
28: }
29:
# Compiler # Resources # Compile Log # Debug # Find Results # Close
<Debug Add watch Next line Continue Next instruction Send command to GDB/next
Line: 6 Col: 5 Insert Done parsing in 0.031 seconds
```

And it executes till line number 27. And you do not get to see any of it; you only see the program running and you should get to see the final result.

(Refer Slide Time: 02:04)

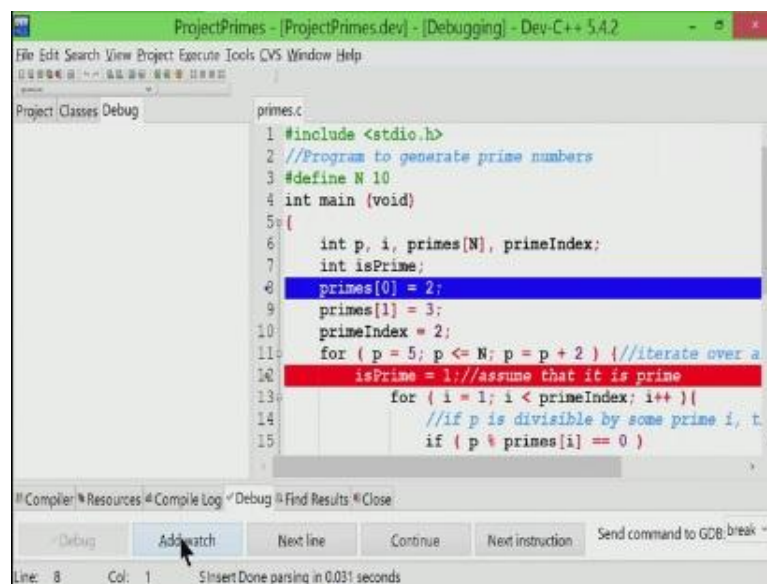


```
ProjectPrimes - [ProjectPrimes.dev] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project: Classes Debug
primes.c
1 #include <stdio.h>
2 //Program to generate prime numbers
3 #define N 10
4 int main (void)
5 {
6     int p, i, primes[N], primeIndex;
7     int isPrime;
8     primes[0] = 2;
9     primes[1] = 3;
10    primeIndex = 2;
11    for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12        isPrime = 1; //assume that it is prime
13        for ( i = 1; i < primeIndex; i++ ){
14            //if p is divisible by some prime i, t
15            if ( p % primes[i] == 0 )

```

But what if I want to see what is the effect of each of these lines? So, I will do single stepping and I can take one line at a time and show the effect of each of these lines; and I will also set up something called a break point, so that I can see how things should run up to the break point. So, let me illustrate those. So, what I am going to do is I am going to set up a break point at line number 8 by pressing on key F4. The moment I press that, you can see that, the background of that line changes to red and there is also a small tick mark on line number 8. I am also going to set up a break point on line number 12. So, I have set it up for break points at eight and 12. So, what is going to happen is – when the program is run, it will not run all the way; first, it will break just before line number 8 and after that if I continue running the program, it will break at line number 12 and so on. So, remember line number 12 is inside the loop. So, it will break; every time, it will give us a chance to inspect values at line number 12. So, to actually look at the values, we need a few other things.

(Refer Slide Time: 03:11)



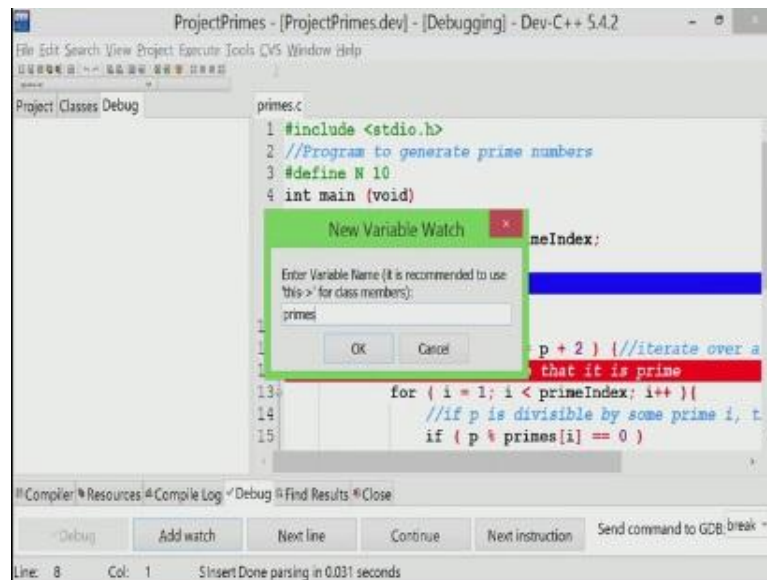
The screenshot shows the Dev-C++ IDE interface. The main window displays the source code for a program named 'primes.c'. The code is as follows:

```
1 #include <stdio.h>
2 //Program to generate prime numbers
3 #define N 10
4 int main (void)
5 {
6     int p, i, primes[N], primeIndex;
7     int isPrime;
8     primes[0] = 2;
9     primes[1] = 3;
10    primeIndex = 2;
11    for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12        isPrime = 1; //assume that it is prime
13        for ( i = 1; i < primeIndex; i++ ) {
14            //if p is divisible by some prime i, t
15            if ( p % primes[i] == 0 )
```

Line 8 is highlighted in blue, and line 12 is highlighted in red. The IDE's status bar at the bottom shows 'Line: 8 Col: 1' and '5 Insert Done parsing in 0.031 seconds'. The 'Debug' menu is open, and the 'Add watch' option is selected.

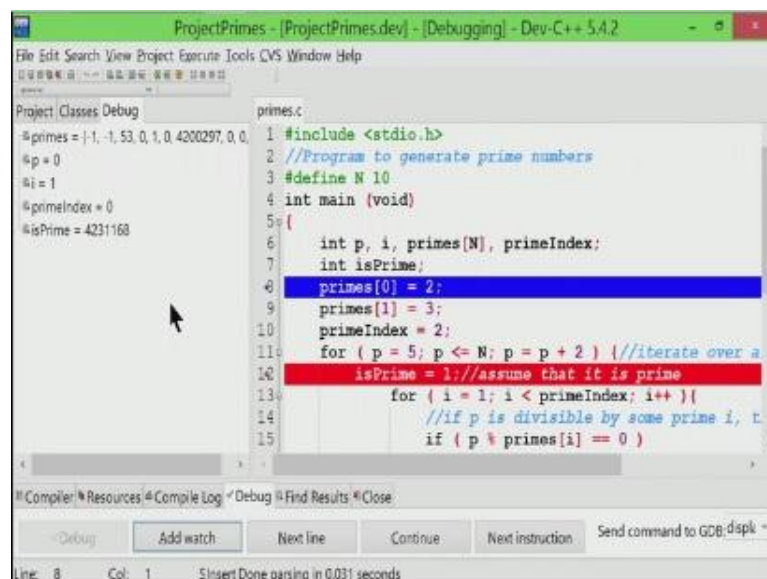
So, I am going to click on debug, so that the debugging is turned on. And I am going to add watch on a few variables.

(Refer Slide Time: 03:20)



So, what are the variables of interest? I want to see what primes is.

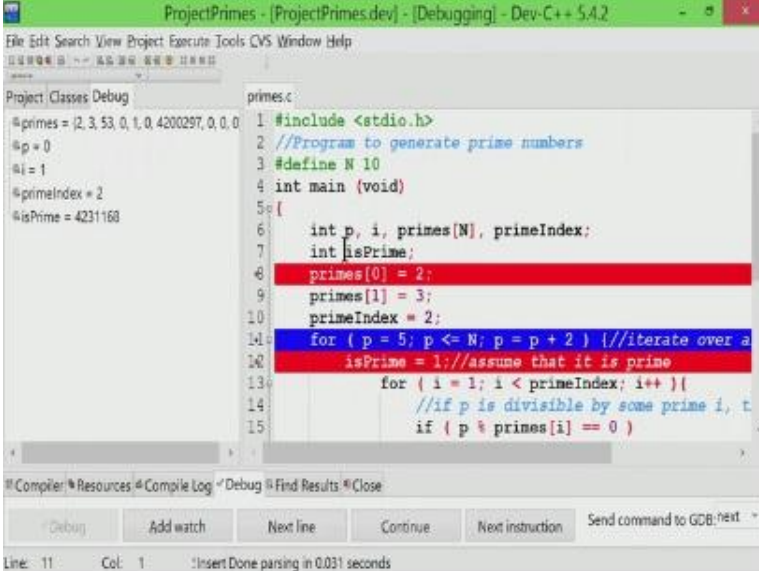
(Refer Slide Time: 03:23)



And if you see that right; so primes seems to have all these values, which we really do not want. And this is the first thing that you should be observing. So, there are values of primes. And since primes is uninitialized, remember we are in... We have still not executed the program; we are just in line number 8. So, we do not know what values must be there for primes. Let us also watch p; which is the value that we want. We watch i. Let us watch prime index. And finally, let us also watch isPrime. So, I want to show

how these things change as we run the program. So, watch the left side; you will see all these variables that are there.

(Refer Slide Time: 04:13)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
@primes = {2, 3, 53, 0, 1, 0, 4200297, 0, 0, 0}
@p = 0
@i = 1
@primeIndex = 2
@isPrime = 4231160
primes.c
1 #include <stdio.h>
2 //Program to generate prime numbers
3 #define N 10
4 int main (void)
5 {
6     int p, i, primes[N], primeIndex;
7     int isPrime;
8     primes[0] = 2;
9     primes[1] = 3;
10    primeIndex = 2;
11    for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12        isPrime = 1; //assume that it is prime
13        for ( i = 1; i < primeIndex; i++ ){
14            //if p is divisible by some prime i, t
15            if ( p % primes[i] == 0 )
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 11 Col: 1 !Insert Done parsing in 0.031 seconds
```

And let see what happens now. So, as of now, I am going to do F7, which is single step. So, it will go one line at a time. So, primes of 0 is now 2 and primes of 1 is 3. So, as I keep pressing F7, it goes one line at a time and whatever line is to be executed will be highlighted. So, as of now, line numbers 8 and 9 have executed. So, 2 and 3 went to the first entries in primes. So, now, I go to line number 10; I execute it. So, prime index became 2. So, I am going to start filling up prime numbers from location 2 onwards. So, as of now, 2 and 3 are correct; I am going to start filling up from location 2 onwards.

(Refer Slide Time: 05:03)

```
1 #include <stdio.h>
2 //Program to generate prime numbers
3 #define N 10
4 int main (void)
5 {
6     int p, i, primes[N], primeIndex;
7     int isPrime;
8     primes[0] = 2;
9     primes[1] = 3;
10    primeIndex = 2;
11    for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12        isPrime = 1; //assume that it is prime
13        for ( i = 1; i < primeIndex; i++ ){
14            //if p is divisible by some prime i, t
15            if ( p % primes[i] == 0 )
```

And what is the value now? p is 5; I am going to check whether 5 is prime or not. I assume that, it is prime to start with. So, now, the whole program is ready for running. We have various things; 2 and 3 are already in place; I know p equal to 5; I have to start checking from 1 onwards and see if it is divisible or not, and if so we are going to see if this is Prime changes.

(Refer Slide Time: 05:31)

```
7     int isPrime;
8     primes[0] = 2;
9     primes[1] = 3;
10    primeIndex = 2;
11    for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12        isPrime = 1; //assume that it is prime
13        for ( i = 1; i < primeIndex; i++ ){
14            //if p is divisible by some prime i, t
15            if ( p % primes[i] == 0 )
16                isPrime = 0;
17        }
18        if ( isPrime == 1 ) {
19            primes[primeIndex] = p;
20            ++primeIndex;
21        }
```

So, let us start moving. i is 1. So, the first... I am at line number 15; p at this point is 5; primes of i. So, primes of 1 is 3. So, this line is checking if 5 percentage 3 is 0 or not. It

is not 0. So, this line will not execute. And let us look at prime index; prime index became 2.

(Refer Slide Time: 06:05)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project: Classes Debug
@primes = {2, 3, 53, 0, 1, 0, 4200297, 0, 0, 0}
@p = 5
@i = 2
@primeIndex = 2
@isPrime = 1
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13 for ( i = 1; i < primeIndex; i++ ) {
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16 isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19 primes[primeIndex] = p;
20 ++primeIndex;
21 }
}
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 19 Col: 1 Insert Done parsing in 0.031 seconds
```

And i is also 2. So, you go and check if this is prime or not. The flag has not changed. So, 5 is a prime number; you record that. So, if... Once I finish line number 19, you will see that, this location – this 53 will change to 5.

(Refer Slide Time: 06:25)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project: Classes Debug
@primes = {2, 3, 5, 0, 1, 0, 4200297, 0, 0, 0}
@p = 5
@i = 2
@primeIndex = 2
@isPrime = 1
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13 for ( i = 1; i < primeIndex; i++ ) {
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16 isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19 primes[primeIndex] = p;
20 ++primeIndex;
21 }
}
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 20 Col: 1 Insert Done parsing in 0.031 seconds
```

We can see that now.

(Refer Slide time: 06:28)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 0, 1, 0, 4200297, 0, 0, 0}
p = 5
i = 2
primeIndex = 3
isPrime = 1
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13 for ( i = 1; i < primeIndex; i++ ) {
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16 isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19 primes[primeIndex] = p;
20 ++primeIndex;
21 }
Compiler Resources Compile Log Debug Find Results Close
-Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 11 Col: 1 !Insert Done parsing in 0.031 seconds
```

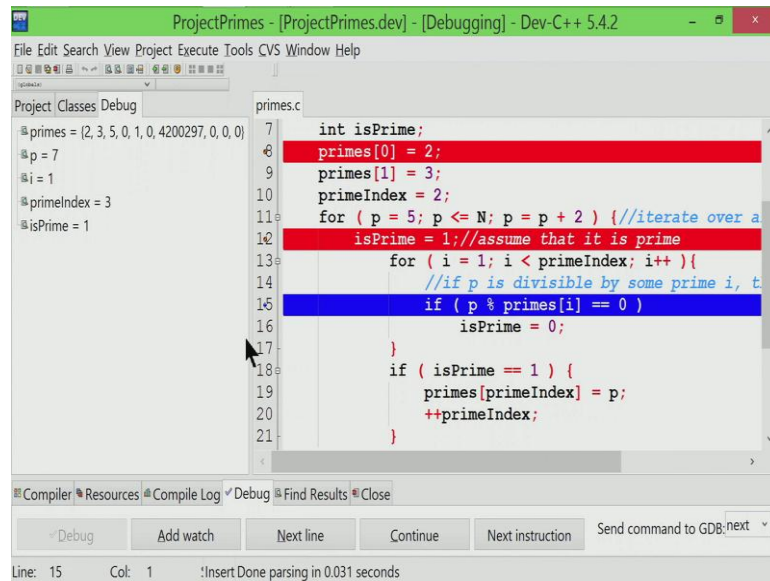
And prime index is 3. So, now, we are ready to see whether the next odd number, which is 7 is prime or not. So, let us continue now.

(Refer Slide Time: 06:38)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 0, 1, 0, 4200297, 0, 0, 0}
p = 7
i = 2
primeIndex = 3
isPrime = 1
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13 for ( i = 1; i < primeIndex; i++ ) {
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16 isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19 primes[primeIndex] = p;
20 ++primeIndex;
21 }
Compiler Resources Compile Log Debug Find Results Close
-Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 13 Col: 1 !Insert Done parsing in 0.031 seconds
```

So, we are ready to check whether 7 is prime or not. We go back and assert 7 is prime. And I am going to check from 1 to i less than prime index. So, prime index is 3. So, I am going to check 7 against 3 and 5.

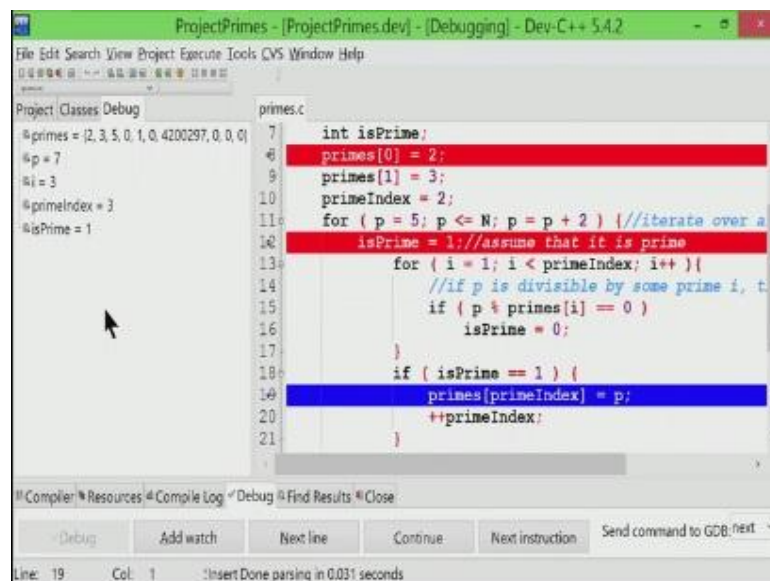
(Refer Slide time: 06:56)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 0, 1, 0, 4200297, 0, 0, 0}
p = 7
i = 1
primeIndex = 3
isPrime = 1
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
}
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 15 Col: 1 :Insert Done parsing in 0.031 seconds
```

So, first, you check 7 against 3; it is not 0.

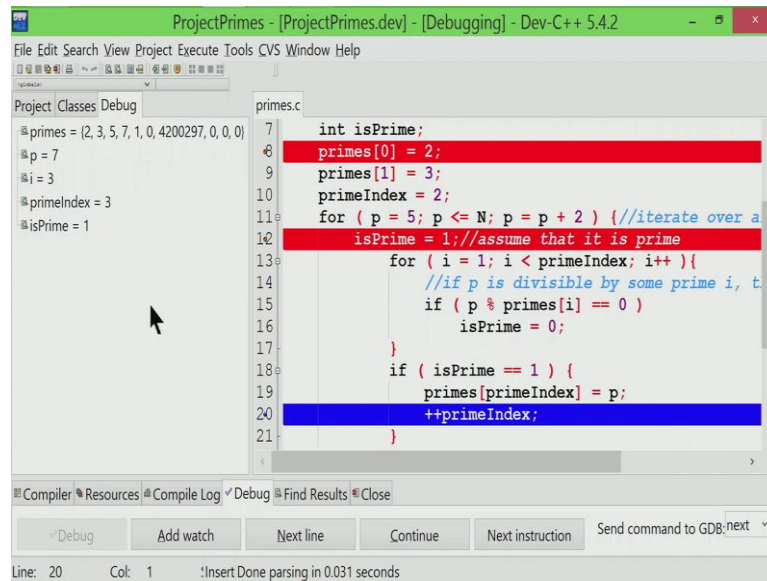
(Refer Slide Time: 07:02)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 0, 1, 0, 4200297, 0, 0, 0}
p = 7
i = 3
primeIndex = 3
isPrime = 1
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
}
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 19 Col: 1 :Insert Done parsing in 0.031 seconds
```

Then, you check 7 against 5; it is also not 0. So, its prime does not change. So, its prime remains at 1.

(Refer Slide Time: 07:12)

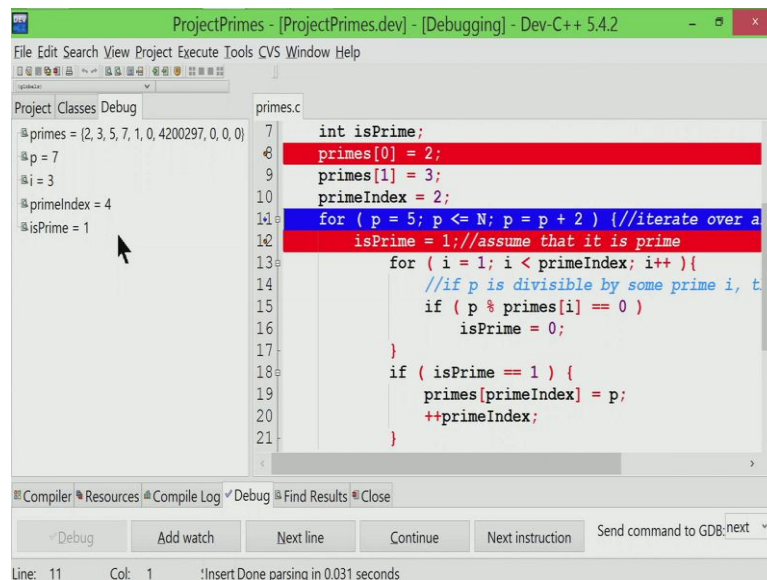


```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 7
i = 3
primeIndex = 3
isPrime = 1
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }

```

So, primes of prime index is – you can record 7; you will see a change here; it changes to 7.

(Refer Slide Time: 07:15)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 7
i = 3
primeIndex = 4
isPrime = 1
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }

```

And prime index will change to 4. So, we are trying to find out the next prime number. Then...

(Refer Slide time: 07:22)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 9
i = 3
primeIndex = 4
isPrime = 1
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13 for ( i = 1; i < primeIndex; i++ ){
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16 isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19 primes[primeIndex] = p;
20 ++primeIndex;
21 }
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 12 Col: 1 !Insert Done parsing in 0.031 seconds
```

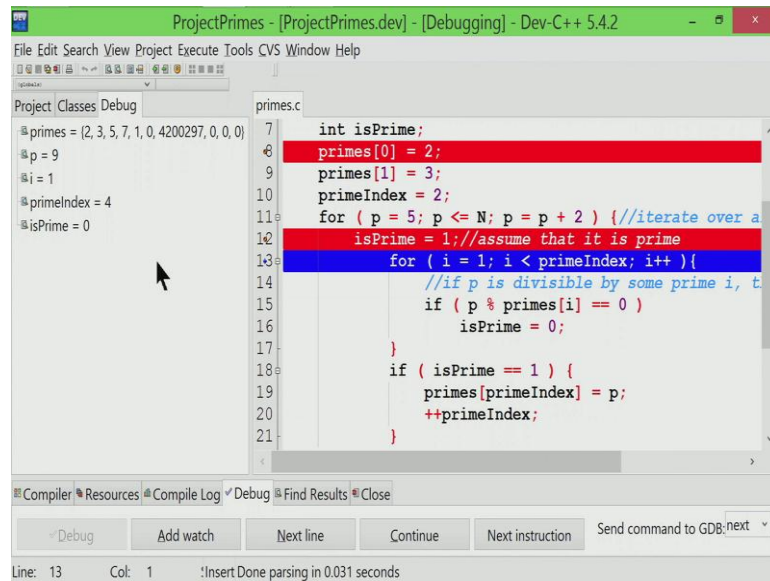
Now, we are checking 9. We assume that, 9 is prime to start with. Then I am going to check against all the numbers from 3 to 7; I am going to check against 9.

(Refer Slide Time: 07:33)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 9
i = 1
primeIndex = 4
isPrime = 1
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13 for ( i = 1; i < primeIndex; i++ ){
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16 isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19 primes[primeIndex] = p;
20 ++primeIndex;
21 }
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 15 Col: 1 !Insert Done parsing in 0.031 seconds
```

If p percentage primes i equal to 0; so 3; 9 percentage 3 is actually 0. So, this is the first time, this line is executed.

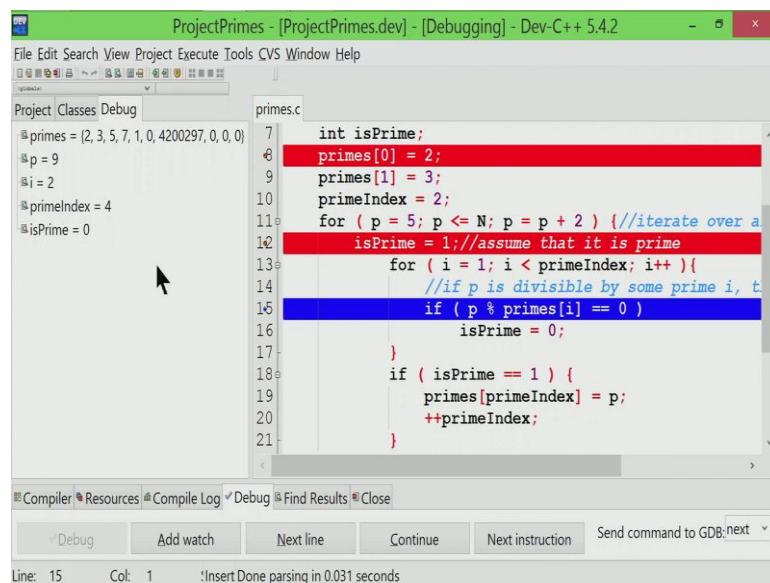
(Refer Slide Time: 07:44)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 9
i = 1
primeIndex = 4
isPrime = 0
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13 for ( i = 1; i < primeIndex; i++ ){
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16 isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19 primes[primeIndex] = p;
20 ++primeIndex;
21 }
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 13 Col: 1 !Insert Done parsing in 0.031 seconds
```

So, isPrime become 0. As I said earlier, we are checking still against 5 and 7.

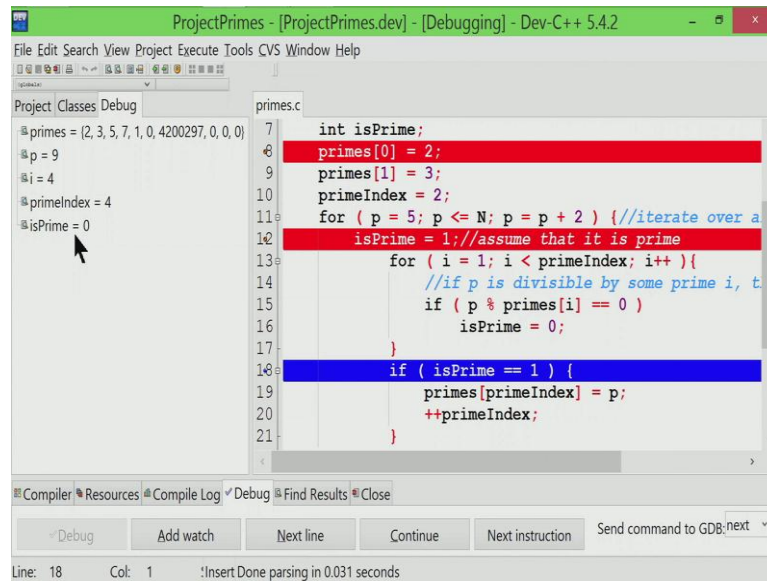
(Refer Slide Time: 07:48)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 9
i = 2
primeIndex = 4
isPrime = 0
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13 for ( i = 1; i < primeIndex; i++ ){
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16 isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19 primes[primeIndex] = p;
20 ++primeIndex;
21 }
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 15 Col: 1 !Insert Done parsing in 0.031 seconds
```

So, we check against 5; then we check against 7.

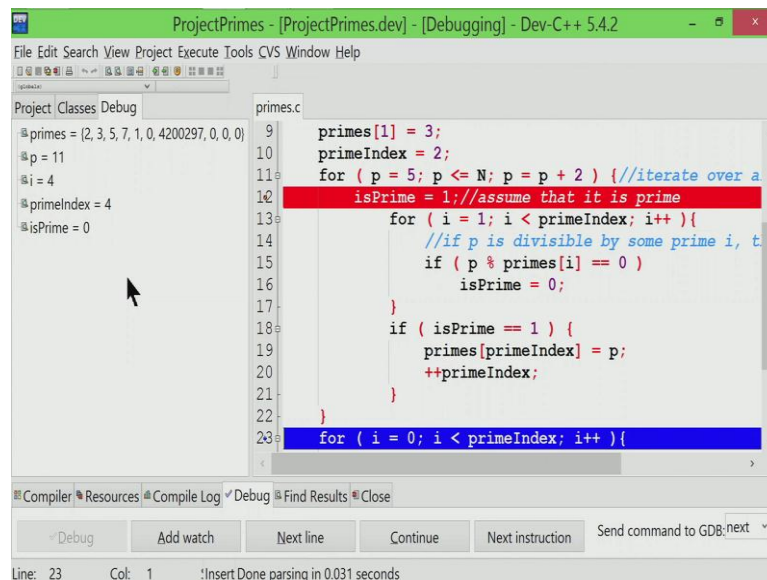
(Refer Slide Time: 07:55)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 9
i = 4
primeIndex = 4
isPrime = 0
primes.c
7 int isPrime;
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13 for ( i = 1; i < primeIndex; i++ ){
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16 isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19 primes[primeIndex] = p;
20 ++primeIndex;
21 }
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 18 Col: 1 !Insert Done parsing in 0.031 seconds
```

And the key thing is isPrime is 0 at this point. So, since isPrime is 0, this block of code from 18 to 20 will not execute; and we are going to now go and check the next number.

(Refer Slide Time: 08:09)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 11
i = 4
primeIndex = 4
isPrime = 0
primes.c
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13 for ( i = 1; i < primeIndex; i++ ){
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16 isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19 primes[primeIndex] = p;
20 ++primeIndex;
21 }
22 }
23 for ( i = 0; i < primeIndex; i++ ){
24
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 23 Col: 1 !Insert Done parsing in 0.031 seconds
```

So, p is 11. So, 11 is actually greater than 10. So, at this point, you exit out of this loop from 12 to 22. Now, ready to print all the prime numbers. So, at this point, you can see that, prime index is 4; which means the first four entries starting from 0-th location till third location have valid prime numbers. So, that is what we are going to do.

(Refer Slide Time: 08:35)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 11
i = 0
primeIndex = 4
isPrime = 0
primes.c
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
22     primeIndex = 4
23     for ( i = 0; i < primeIndex; i++ ){
24         printf ("%d ", primes[i]);
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 24 Col: 1 !Insert Done parsing in 0.031 seconds
```

We can check and print prime numbers.

(Refer Slide Time: 08:37)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 11
i = 1
primeIndex = 4
isPrime = 0
primes.c
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
22     }
23     for ( i = 0; i < primeIndex; i++ ){
24         printf ("%d ", primes[i]);
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 24 Col: 1 !Insert Done parsing in 0.031 seconds
```

So, print 2.

(Refer Slide Time: 08:39)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 11
i = 2
primeIndex = 4
isPrime = 0
primes.c
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
22 }
23 for ( i = 0; i < primeIndex; i++ ){
24     printf ("%d ", primes[i]);
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 24 Col: 1 !Insert Done parsing in 0.031 seconds
```

Print 3

(Refer Slide Time: 08:40)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 11
i = 3
primeIndex = 4
isPrime = 0
primes.c
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
22 }
23 for ( i = 0; i < primeIndex; i++ ){
24     printf ("%d ", primes[i]);
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 24 Col: 1 !Insert Done parsing in 0.031 seconds
```

Print 5.

(Refer Slide Time: 08:41)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
@primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
@p = 11
@i = 3
@primeIndex = 4
@isPrime = 0
primes.c
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
22 }
23 for ( i = 0; i < primeIndex; i++){
24     printf ("%d ", primes[i]);
25 }
26 printf ("\n");
27 return 0;
28
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 23 Col: 1 Insert Done parsing in 0.031 seconds
```

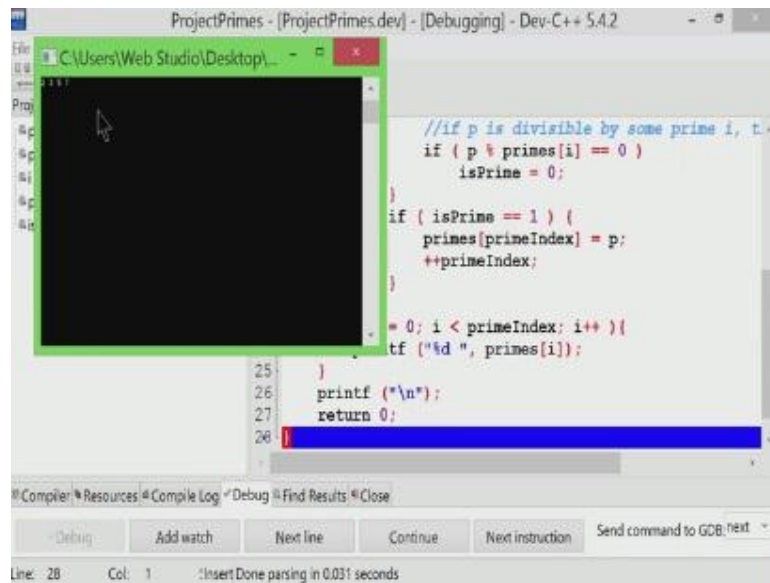
Print 7.

(Refer Slide Time: 08:42)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
@primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
@p = 11
@i = 4
@primeIndex = 4
@isPrime = 0
primes.c
14 //if p is divisible by some prime i, t
15 if ( p % primes[i] == 0 )
16     isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19     primes[primeIndex] = p;
20     ++primeIndex;
21 }
22 }
23 for ( i = 0; i < primeIndex; i++){
24     printf ("%d ", primes[i]);
25 }
26 printf ("\n");
27 return 0;
28
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:next
Line: 26 Col: 1 Insert Done parsing in 0.031 seconds
```

And we are done. So, at this point, we are done.

(Refer Slide Time: 08:50)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
C:\Users\Web Studio\Desktop\...
//if p is divisible by some prime i, t-
if ( p % primes[i] == 0 )
    isPrime = 0;
}
if ( isPrime == 1 ) {
    primes[primeIndex] = p;
    ++primeIndex;
}
= 0; i < primeIndex; i++){
    printf ("%d ", primes[i]);
}
25 }
26 printf ("\n");
27 return 0;
28
```

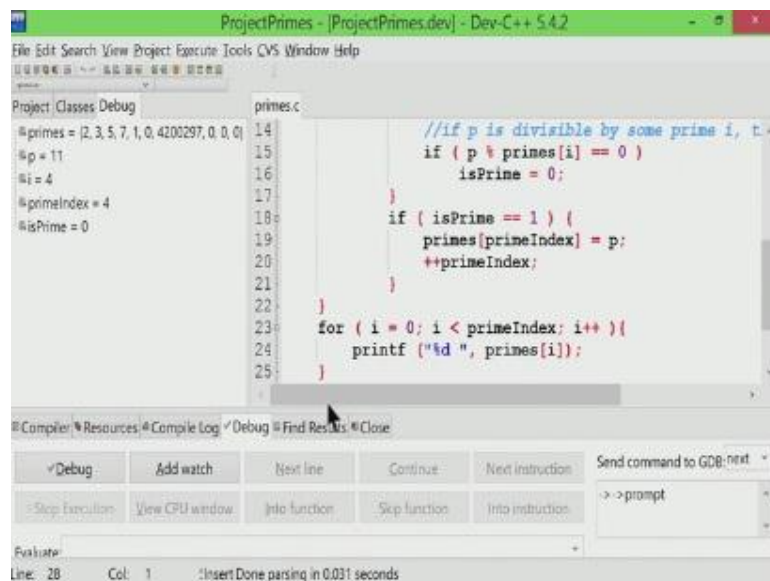
Compiler Resources Compile Log Debug Find Results Close

Debug Add watch Next line Continue Next instruction Send command to GDB:next

Line: 28 Col: 1 :Insert Done parsing in 0.031 seconds

So, if we go and look at this, we already see 2, 3, 5 and 7 printed.

(Refer Slide Time: 08:59)



```
ProjectPrimes - [ProjectPrimes.dev] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
@primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
@p = 11
@i = 4
@primeIndex = 4
@isPrime = 0
primes.c
14 //if p is divisible by some prime i, t-
15 if ( p % primes[i] == 0 )
16     isPrime = 0;
17 }
18 if ( isPrime == 1 ) {
19     primes[primeIndex] = p;
20     ++primeIndex;
21 }
22 }
23 for ( i = 0; i < primeIndex; i++){
24     printf ("%d ", primes[i]);
25 }
26 }
27 return 0;
28
```

Compiler Resources Compile Log Debug Find Results Close

Debug Add watch Next line Continue Next instruction Send command to GDB:next

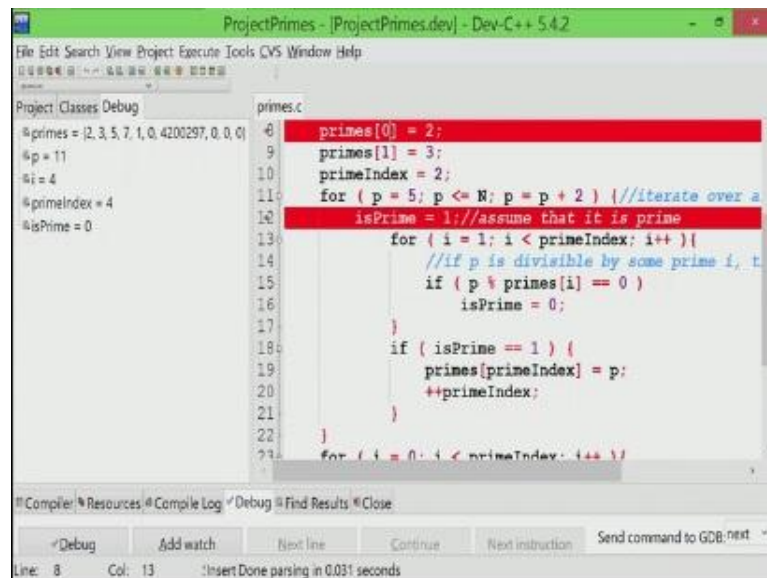
Stop execution View CPU window Into function Step function Into instruction

Evaluate

Line: 28 Col: 1 :Insert Done parsing in 0.031 seconds

And once this is all over, you can stop the execution. So, this program is actually correct. I can stop the execution and so on.

(Refer Slide Time: 09:03)



The screenshot shows a C++ IDE window titled "ProjectPrimes - [ProjectPrimes.dev] - Dev-C++ 5.4.2". The main editor displays the source code for "primes.c". The code is as follows:

```
08 primes[0] = 2;
09 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
22 }
23 for ( i = 0; i < primeIndex; i++ ) {
```

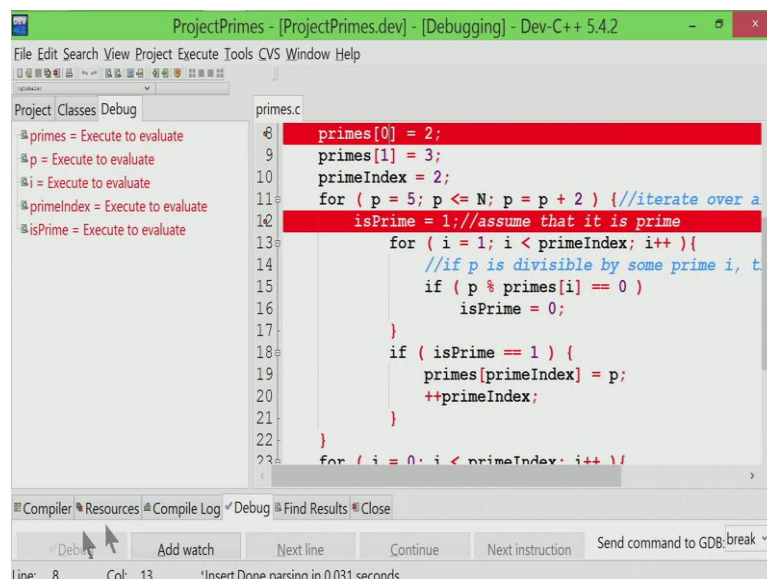
The debugger window on the left shows the following state:

- Project: Classes Debug
- primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
- p = 11
- i = 4
- primeIndex = 4
- isPrime = 0

The status bar at the bottom indicates "Line: 8 Col: 13 !Insert Done parsing in 0.031 seconds".

So, what you are really seeing is this break point gives you an opportunity to start from a particular. So, let us say these break points 8 and 12. If I start at 8, it actually gives me an opportunity to run till 12 without executing the lines in between. Even though I was doing single stepping, I actually showed every single statement and how it was executing. You do not really have to do that. So, let me quickly demonstrate that.

(Refer Slide Time: 09:33)



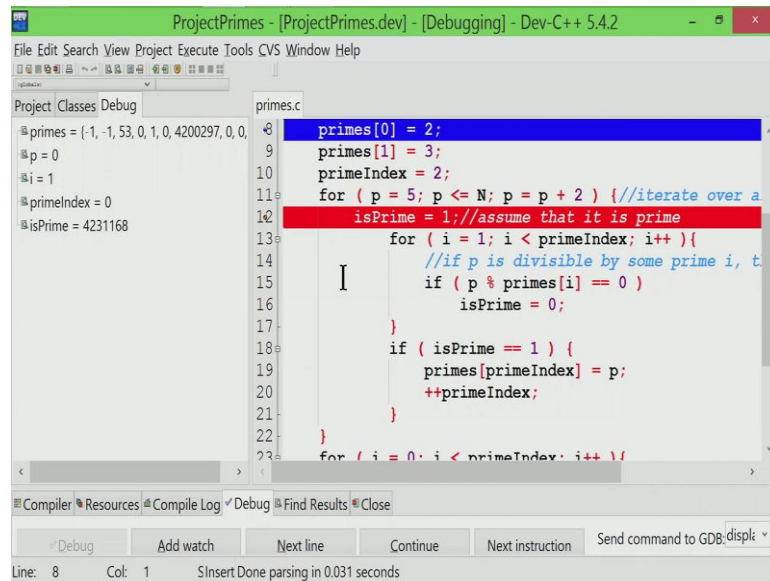
The screenshot shows the same C++ IDE window, but now in a debugging state. The title bar includes "[Debugging]". The source code is the same as in the previous screenshot. The debugger window on the left now lists several variables to be evaluated:

- primes = Execute to evaluate
- p = Execute to evaluate
- i = Execute to evaluate
- primeIndex = Execute to evaluate
- isPrime = Execute to evaluate

The status bar at the bottom indicates "Line: 8 Col: 13 !Insert Done parsing in 0.031 seconds".

I am going to run one more debug now.

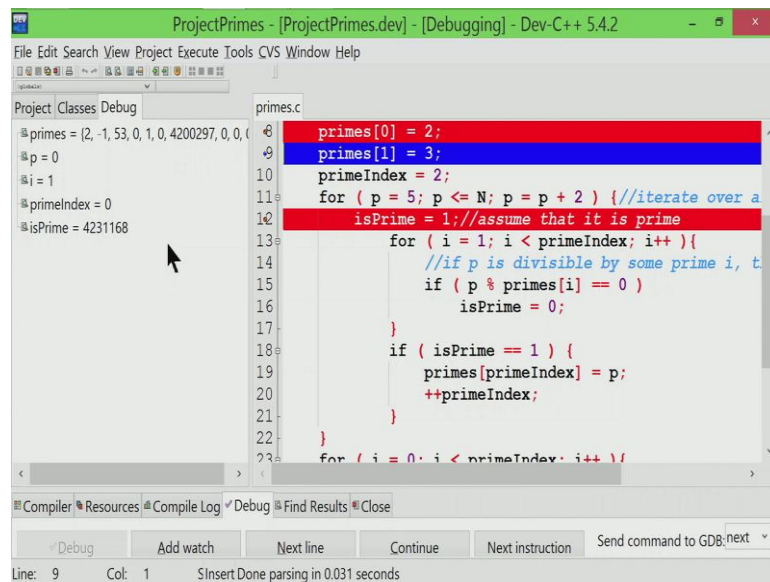
(Refer Slide Time: 09:34)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = [-1, -1, 53, 0, 1, 0, 4200297, 0, 0]
p = 0
i = 1
primeIndex = 0
isPrime = 4231168
primes.c
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12     isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ) {
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
22 }
23 for ( i = 0; i < primeIndex; i++ ) {
24     //if p is divisible by some prime i, t
25     if ( p % primes[i] == 0 )
26         isPrime = 0;
27 }
28 }
29 }
30 }
31 }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
```

Again, I am watching all these things. You can see that, all these prime numbers and so on are invalid.

(Refer Slide Time: 09:42)



```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = [2, -1, 53, 0, 1, 0, 4200297, 0, 0]
p = 0
i = 1
primeIndex = 0
isPrime = 4231168
primes.c
9 primes[0] = 2;
10 primes[1] = 3;
11 primeIndex = 2;
12 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
13     isPrime = 1; //assume that it is prime
14     for ( i = 1; i < primeIndex; i++ ) {
15         //if p is divisible by some prime i, t
16         if ( p % primes[i] == 0 )
17             isPrime = 0;
18     }
19     if ( isPrime == 1 ) {
20         primes[primeIndex] = p;
21         ++primeIndex;
22     }
23 }
24 for ( i = 0; i < primeIndex; i++ ) {
25     //if p is divisible by some prime i, t
26     if ( p % primes[i] == 0 )
27         isPrime = 0;
28 }
29 }
30 }
31 }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
```

So, I start running the program. So, primes of 0 is 2.

(Refer Slide Time: 09:48)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 53, 0, 1, 0, 4200297, 0, 0, 0}
p = 5
i = 1
primeIndex = 2
isPrime = 4231168
primes.c
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
22 }
23 for ( i = 0; i < primeIndex; i++ ) {
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:cont
Line: 12 Col: 1 !Insert Done parsing in 0.031 seconds Continue
```

Then, I can press continue; it will take me to the next break point. So, at this point, it went from line number 8 to 12 directly without showing each of these steps. So, now, since the next break point is 12 and it is inside the loop from line number 11 to 22; if I click on continue, we will do everything that is required for the loop and go to the next time when the p itself is changing.

(Refer Slide Time: 10:16)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 0, 1, 0, 4200297, 0, 0, 0}
p = 7
i = 2
primeIndex = 3
isPrime = 1
primes.c
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13     for ( i = 1; i < primeIndex; i++ ){
14         //if p is divisible by some prime i, t
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
22 }
23 for ( i = 0; i < primeIndex; i++ ) {
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:cont
Line: 12 Col: 1 !Insert Done parsing in 0.031 seconds Continue
```

So, p changes to 7. You can see the left side; p changes to 7;

(Refer Slide Time: 10:19)

```
ProjectPrimes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 9
i = 3
primeIndex = 4
isPrime = 1
primes.c
8 primes[0] = 2;
9 primes[1] = 3;
10 primeIndex = 2;
11 for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12 isPrime = 1; //assume that it is prime
13
14     for ( i = 1; i < primeIndex; i++ ) {
15         //if p is divisible by some prime i, t
16         if ( p % primes[i] == 0 )
17             isPrime = 0;
18     }
19     if ( isPrime == 1 ) {
20         primes[primeIndex] = p;
21         ++primeIndex;
22     }
23 }
24 for ( i = 0; i < primeIndex; i++ ) {
25     printf("%d ", primes[i]);
26     if ( i % 10 == 9 ) printf("\n");
27 }
28 }
```

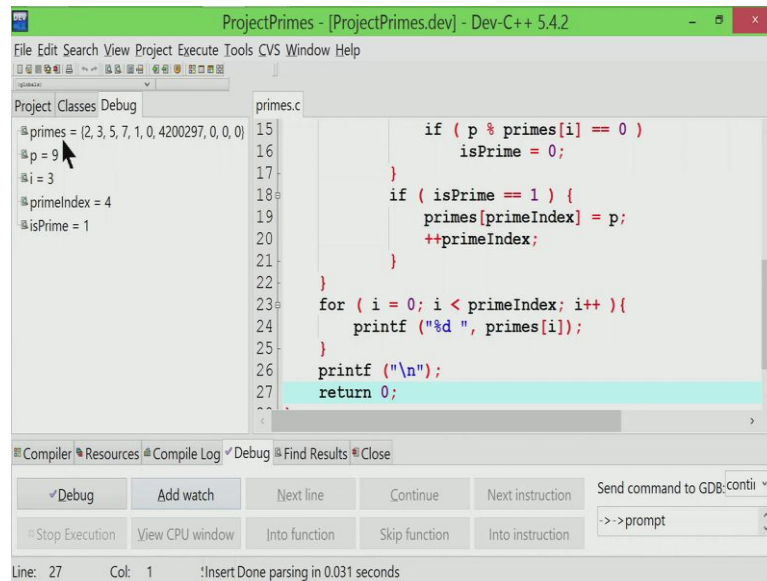
p changes to 9 and so on. So, all the things that were happening here; all the work got done; it is not that, the work did not get done; the work got done; but you are waiting at line number 12 and seeing what is happening.

(Refer Slide Time: 10:35)

```
ProjectPrimes - [ProjectPrimes.dev] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 9
i = 3
primeIndex = 4
isPrime = 1
primes.c
1 #include <stdio.h>
2 //Program to generate prime numbers
3 #define N 10
4 int main (void)
5 {
6     int p, i, primes[N], primeIndex;
7     int isPrime;
8     primes[0] = 2;
9     primes[1] = 3;
10    primeIndex = 2;
11    for ( p = 5; p <= N; p = p + 2 ) { //iterate over a
12    isPrime = 1; //assume that it is prime
13
14        for ( i = 1; i < primeIndex; i++ ) {
15            //if p is divisible by some prime i, t
16            if ( p % primes[i] == 0 )
17                isPrime = 0;
18        }
19        if ( isPrime == 1 ) {
20            primes[primeIndex] = p;
21            ++primeIndex;
22        }
23    }
24    for ( i = 0; i < primeIndex; i++ ) {
25        printf("%d ", primes[i]);
26        if ( i % 10 == 9 ) printf("\n");
27    }
28 }
```

So, at this point prime is... You are still checking for 9. And the whole thing got done now.

(Refer Slide Time: 10:43)



```
ProjectPrimes - [ProjectPrimes.dev] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
Project Classes Debug
primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}
p = 9
i = 3
primeIndex = 4
isPrime = 1
primes.c
15         if ( p % primes[i] == 0 )
16             isPrime = 0;
17     }
18     if ( isPrime == 1 ) {
19         primes[primeIndex] = p;
20         ++primeIndex;
21     }
22 }
23 for ( i = 0; i < primeIndex; i++ ){
24     printf ("%d ", primes[i]);
25 }
26 printf ("\n");
27 return 0;
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction Send command to GDB:Conti
Stop Execution View CPU window Into function Skip function Into instruction ->>prompt
Line: 27 Col: 1 !Insert Done parsing in 0.031 seconds
```

You can go and see that, the program is executed. And if I... We have run till the end of the program. So, at this point, you can stop the execution. And all the values from 2, 3 up to 7 are already in the primes array. So, you can use the debugger for debugging various programs. This is a very effective tool. I suggest that you get used to this debugging, so that you can check all your programs once before you go and do your home works on the left side. And this is a very useful utility. So, I cannot emphasize this more. You have a good handle of how the debugger is used. And you do not have to print screen – fulls of debugging statements; instead, use the debugger effectively. So, the key trick will be in finding out what variables you want to watch and where to set your break points. So, I knew this ahead of times. So, I had set break points at line number 12 and line number 8. But, you have to be careful about where you are setting the break points. It does not make sense to set break points at every line.

Thank you very much.