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<=N</li> 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 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)

	P	rojectPrimes - (P	rojectPrimes.dev]	- Dev-C++ 542	- 8 - 4
He Edit Search View	e Project Execute IA are sa a anta ug	ools CVS Window is primes.c 1 #inclus 2 //Prog 3 #defins 4 int ma: 5 c ( 6 int 7 int 8 pr: 9 pr: 10 pr: 11 fo: 12 13 14 15	<pre>idp is <stdio.h> ram to generat a N 10 in (void) t p, i, primes t isPrime; imes[0] = 2; imes[1] = 3; imeIndex = 2; r ( p = 5; p &lt;     isPrime = 1;     for ( i         //if         if (</stdio.h></pre>	<pre>e prime number [N], primeInde = N; p = p + 2 //assume that = 1; i &lt; prime p is divisibl p % primes[i]</pre>	<pre>x; () {//iterate over a it is prime Index; i++ ){ e by some prime i, t == 0 } .</pre>
Compiler Resource	ces #Compile Log *	Debug R Find Result	ts ¶Close		
✓Debug	Add watch	Nextine	Continue	Next instruction	Send command to GDB:Next *
Line: 7 Col	: 17 : :Insert	Done parsing in 0.0	B1 seconds		

We saw this program earlier.

(Refer Slide Time: 01:04)

	P	rojectPrimes -	ProjectPrim	es.dev] -	Dev-C++ 5.4.2	- 3	5 X
File Edit Search View	Project Execute Io	ools CVS Windo	w Help				
Project Classes Deb	ug	primes.c					
		3 #def 4 int : 5 { 6 7 8 1 9 1 10 1 12 13 14 15 16 17	<pre>ine N 10 main (void) int p, i, p int isPrime primes[0] = primes[1] * primeIndex for ( p = 1</pre>	<pre>&gt;rimes[1 % ] = 2; = 3; = 2; 5; p &lt;= = 1;/, r (i = //if ; if ( ) if ( )</pre>	N; p = p + 2 /assume that 1; i < prime p is divisibl p % primes[i] SPrime = 0;	<pre>x; ) (//iterate o it is prime Index; i++ )( e by some prim == 0 )</pre>	over a
Compiler Resource	ces 4 Compile Log 🗸	Debug 4 Find Re	isults @Close				
"Debug	Add watch	Next in	e Con	tinue:	Next instruction	Send command to G	DB.next -
Line: 3 Col	: 13 :lisert	Done parsing in	0.031 seconds				

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	- Projec	(Primes.dev	- Dev-C++ 5.4.2	- 0 ×
File Edit Search Vie District and the search vie The search of the search vie Project Classes Deb	W Droject Execute In the set of t	Project Primes Teols CVS Wind primes.c 15 16 17 18 19 20 21 22 23 24 25 26 27	<pre>. [Projec ow Help for ( : printf return</pre>	<pre>Phrmes.dev] if ( if (isP prim ++pr ) i = 0; i &lt; intf ("%d (*\n*); 0;</pre>	<pre>- Dev-C++ 5.42 p % primes[i] isPrime = 0; rime == 1 ) { es[primeIndex] imeIndex; primeIndex; i ", primes[i]);</pre>	0 ) 
	28 } 29					
I Compiler Resour	rces #Compile Log	"Debug # Find #	lesuits #Ck	ose		
- Debug	Add watch	Next in	e	Continue	Next instruction	Send command to GDB: next *
Line: 6 Co	k: 5 Sins	ert Done parsing i	n 0.031 seo	onds		

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)



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)



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)

Project Classes Debu	99 99	primes.	: include <	stdio.h>				
		2 // 3 #c 4 in	/Program define N nt main (	to generate 10 void)	e prime numb	ers		
		Ent the	New V er Variable Nam s >' for dass me mes	ariable Watch e (it is recommende mbers):	alto use	dex;		
			OK	Cancel	p +	2 ) (//ite t it is pr	erate over a	
		130 14 15		for ( i * //if if (	1; i < pri p is divisi p prines[	<pre>meIndex; i+ ble by some i] == 0 )</pre>	+)( prime i, t	
Il Compiler * Resourc	es #Compile Log *	Debug A fir	nd Results #Cl	ose			*	
	Add watch	Nex	tline	Continue	Next instruction	Send comm	and to GDB break	•

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

(Refer Slide Time: 03:23)

ProjectPi	imes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2 - 🔍
Ble Edit Search View Broject Execute I	cols QVS Window Help
Project (Classes Debug * primes = (-1, -1, 53, 0, 1, 0, 4200297, 0 * p = 0 * i = 1 * primeIndex + 0 * isPrime = 4231168	<pre>primes: 0 1 finclude <stdio.h> 2 //Program to generate prime numbers 3 fdefine 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 primes[1] = 3; 11 for ( p = 5; p &lt;= N; p = p + 2 ) {//iterate over a 12 isPrime = 1://assume that it is prime 13 for ( i = 1; i &lt; primeIndex; i++ ){ 14 //if p is divisible by some prime 1, t 15 if ( p * primes[1] == 0 ) 14 } 15 } 15 } 16 } 17 \$ 18 \$ 19 \$ 19 \$ 10 \$ 10 \$ 10 \$ 10 \$ 10 \$ 10 \$ 10 \$ 10</stdio.h></pre>
4	3 ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) (
Compiler Resources Compile Log	Debug % Find Results * Close
<group add="" td="" watch<=""><td>Next line Continue Next instruction Send command to GDB/disple =</td></group>	Next line Continue Next instruction Send command to GDB/disple =
Line: 8 Col: 1 Sinser	t Done parsing in 0.031 seconds

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)

	ProjectPrime	s - (ProjectPrime	es.dev] - [Debu	gging] - Dev-C++	5.4.2 - a ×
File Sdt Search View Proje Search View Proje Search View Proje Project (Classes Debug Sprimes = (2, 3, 53, 0, 1, 0, Sprimes = (2, 3, 53, 0, 1, 0,	4200297, 0, 0, 0	<pre>s [ProjectPrime CVS Window Help primes.c 1 #include 2 //Program 3 #define N 4 int main 5 c [ 6 int p 7 int <u>p</u> 9 prime 9 prime 10 prime 10 for ( 4 1 10 for ( 10 for ( 10</pre>	<pre>cstdio.h&gt; to generat: 10 (void) , i, primes sprime; s[0] = 2; s[1] = 3; Index = 2; p = 5; p &lt; sprime = 1; for ( 1; //if</pre>	<pre>ggmg] - Dev-C++ e prime number [N], primeInde [N], p = p + 2 //assume that l; i &lt; prime p is divisibl</pre>	<pre>x; } //iterate over a it is primo Index; i++ )( e by some prime 1, t</pre>
		15	if (	p % primes[i]	== 0 ) .
GI	- B3				2
Il Compiler * Resources # C	ompile Log - Debu	ng 🛛 Find Results 🌒	Close		
Cebug J	Add watch	Next line	Continue	Next instruction	Send command to GDB: heit *
Line: 11 Col. 1	linsert Don	e parsing in 0.031 se	econds		

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)

ProjectPrim	nes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2 - 🔹 🚺
He filt Search View Project Execute Tool	s CVS Window Help
Project Classes Debug	primes.c
-#primes = (2, 3, 53, 0, 1, 0, 4200297, 0, 0, 0 #p = 5 #i = 1 #primeIndex = 2 #isPrime = 1	<pre>1 finclude <stdio.h> 2 //Program to generate prime numbers 3 #define N 10 4 int main (void) 5 0 { 6 int p, i, primes[N], primeIndex; 7 int isPrime; 6 primes[0] = 2; 9 primes[1] = 3; 10 primeIndex = 2; 11 for ( p = 5; p &lt;= N; p = p + 2 ) {//iterate over a 12 isPrime = 1://assume that it is prime 13 for ( i = 1; i &lt; primeIndex; i++ )[ 14 //if p is divisible by some prime i, t 15 if ( p is primeIndex; i = 0 ) </stdio.h></pre>
1 2	
"Compiler & Resources 4 Compile Log * De	bug © Find Results   Close
Debug Add watch	Next line Continue Next instruction Send command to GDB. heat *
Line: 13 Col: 1 Sinsert D	one parsing in 0.031 seconds

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)



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)

	ProjectPrim	es - [Pro	jectPrime	es.dev] - [Debi	igging] - Dev-C++	542 - e ×
File Solt Search Vie Durage a as Project Classes Deb Sprimes = [2, 3, 53 Sp = 5 Ali = 2 SprimeIndex = 2 SisPrime = 1	* Project Execute Tool 34 94 9 1441 ug 0, 1, 0, 4200297, 0, 0, 0	s CVS Win primes.c 7 6 9 10 11 10 13 14 15 16 17 18 10 20 21	dow Help int i prime prime for (	<pre>sPrime; s[0] = 2; s[1] = 3; Index = 2; p = 5; p &lt; sPrime = 1; for ( i</pre>	<pre>&gt;&gt; N; p = p + 2 //assume that = 1; i &lt; prime 'p is divisibl p % primes[i] isPrime = 0; 'rime == 1 ) { es[primeIndex] </pre>	<pre>i) {//iterate over a it is prime index; i++ ){ ie by score prime i, t == 0 ) </pre>
+	P.					
Il Compiler & Resour	ces #Compile Log * De	bug % Find	Results *	Close		
Debug	Add watch	Next	line	Continue	Next instruction	Send command to GDB:hext *
Line: 19 Co	t Insert Do	one parsing	in 0.031 s	econds		

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)

Proje	ctPrimes - [Pr	ojectPrin	nes.dev] - [Debu	igging] - Dev-C++	542 - 0
Ble Edit Search View Project Eperu I un de a	tr Jools CVS W 10 10 10 10 11 14 15 16 17 18 19 20 21	indow Hel; int : prim prim for	<pre>isPrime; isPrime; es[1] = 3; Endex = 2; ( p = 5; p &lt; 4; for ( i</pre>	<pre>&gt; N; p = p + 2 //assume that &gt; 1; i &lt; prime p is divisibl p is primes[i] isPrime = 0; rime == 1 ) { es[primeIndex]; </pre>	) (//iterate over a it is prime Index; i++ )( e by some prime i, t = 0 ) I = p;
Compiler Resources Compile L	og ≁Debug ¤ Fir	nd Results 4	Close		
-Orbug Add watch	h Neo	d line	Continue	Next instruction	Send command to GDB:next *
Line 20 Col: 1 :1	rsert Done parsi	ng in 0.031	seconds		

We can see that now.

(Refer Slide time: 06:28)

Project Classes Debug	primes.c					
■ primes = {2, 3, 5, 0, 1, 0, 4200297, 0, 0, 0}	7	int isH	Prime;			
i≊p = 5	-8	primes	0] = 2;			
15 j = 2	9	primes	[1] = 3;			
sprimeIndex = 3	10	primeIr	dex = 2;			
isPrime = 1	110	for ( p	) = 5; p <=	N; $p = p + 2$	) {//iterate over a	
	12	isI	<pre>prime = 1;/</pre>	/assume that	it is prime	
	130		for ( i =	1; i < prime	Index; i++ ){	
	14 //if p is divisible by some prime i, t					
	15 if ( p % primes[i] == 0 )					
	16		i	isPrime = 0;		
	17 -		}		Ĩ	
	180		if ( isPr	ime == 1 ) {	1	
	19		prime	s[primeIndex]	= p;	
	20		++pri	meIndex;		
	21 -		}			
					>	
Compiler Sesources Compile Log VDe	bug 🖻 Find	Results	se			
		P	C I	Marking and a	Send command to GDB:next	

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)

<u>File Edit Search View Project Execute Too</u>	ls <u>C</u> VS <u>W</u> in	idow <u>H</u> elp					
Project Classes Debug	primes.c		Duime				
Primes = {2, 3, 5, 0, 1, 0, 4200297, 0, 0, 0}	-8	primes	[0] = 2:				
sp = 7 si - 2	9	primes	[1] = 3;				
BorimeIndex = 3	10	primeI	ndex = 2;				
BisPrime = 1	110	for (	p = 5; p <=	= N; p = p + 2	) {//iterate over	: a.	
-brinne - i	12	is	Prime = 1;	/assume that .	it is prime		
	130		for ( i	= 1; i < prime	Index; i++ ){		
	14	14 //if p is divisible by some prime i, t					
	15		if (	<pre>p % primes[i]</pre>	== 0 )		
	16		i	isPrime = 0;			
	17 -		}				
	180		if ( isPı	rime == 1 ) {			
	19		prime	es[primeIndex]	= p;		
	20		++pri	imeIndex;			
	21		}				
						>	
Compiler Resources Compile Log V De	ebug 🖻 Find	Results C	ose				
Debug     Add watch	Next	line	Continue	Next instruction	Send command to GDB: ne	xt `	

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)

Image: Second	primes.c           7         ii           8         F           9         K           10         F           120         14           15         16           17         180           19         20	<pre>nt isPrime; rimes[0] = 2; rimes[1] = 3; rimeIndex = 2; for ( p = 5; p &lt;&lt;</pre>	<pre>= N; p = p + 2 //assume that = 1; i &lt; prime p is divisibl p % primes[i] isPrime = 0; rime == 1) { es[primeIndex] imeIndex;</pre>	<pre>) {//iterate over a it is prime Index; i++ ){ e by some prime i, t. == 0 ) = p;</pre>
	21	}		>
Compiler Resources Compile Log VD	ebug <sup>©</sup> Find Re	sults Close		
Debug Add watch	Next line	Continue	Next instruction	Send command to GDB: next *

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

(Refer Slide Time: 07:02)

ProjectPrin	es - [Projecti	Primes dev] - (Debu	gging] - Dev-C++	542 - •		
He Edit Search View Project Execute Tool	s 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	<pre>primes: 7     int isPrime; 9     primes[1] = 3; 9     prime[1] = 3; 10     primeIndex = 2; 11     for ( p = 5; p &lt;= N; p = p + 2 ) {//iterate over a 12     isPrime = 1;//assume that if is prime 13     for ( i = 1; i &lt; primeIndex; i++ ){ 14</pre>					
	180 1 <del>0</del> 20 21	if ( isP print ++pri }	rine == 1 ) ( s[prineIndex] imeIndex;	- p) 		
Il Compiler * Resources 4 Compile Log "De	bug <sup>(1)</sup> Find Res	its # Close				
Debug Add watch	Next line	Continue	Next instruction	Send command to GDB: next *		
Line: 19 Col: 1 :Insert D	one 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)

Project Classes Debug	primes.c	
ℙ = (2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0) &p = 7 &i = 3 &primeIndex = 3 &isPrime = 1	7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 -	<pre>int isPrime; primes[0] = 2; primes[1] = 3; primeIndex = 2; for ( p = 5; p &lt;= N; p = p + 2 ) {//iterate over a     isPrime = 1;//assume that it is prime     for ( i = 1; i &lt; primeIndex; i++ ) {         //if p is divisible by some prime i, t         if ( p % primes[i] == 0 )             isPrime = 0;         }         if ( isPrime == 1 ) {             primeIndex; = p;             ++primeIndex;         } } </pre>
≝Compiler ●Resources ≜Compile Log ✓De	bug 🖻 Find I	Results # Close
«Debug Add watch	Nevt li	ine Continue Next instruction Send command to GDB:next *

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)

Project Classes Debug	primes.c							
© primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0} © p = 7 © i = 3 © primeIndex = 4 © isPrime = 1	7 8 9 10 110	<pre>7 int isPrime; 6 primes[0] = 2; 9 primes[1] = 3; 10 primeIndex = 2; 11 for ( p = 5; p &lt;= N; p = p + 2 ) {//iterate over a</pre>						
ł	12 13⊕ 14 15 16 17 - 18⊕	12       isPrime = 1;//assume that it is prime         130       for (i = 1; i < primeIndex; i++) {						
	19 20 21	prim ++pr: }	es[primeIndex] imeIndex;	= p;				
Compiler Resources Compile Log VDe	bug B Find R	esults Close						
Debug Add watch	Next lin	e <u>C</u> ontinue	Next instruction	Send command to GDB: next *				

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

(Refer Slide time: 07:22)

File Edit Search View Project Execute Too	ls <u>C</u> VS <u>W</u> i	ndow <u>H</u> el	p		
Project Classes Debug	primes.c				
-ª primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0 -≋ p = 9 =i = 3 =# primeIndex = 4 =® isPrime = 1	7 6 9 10 11 12 13 14 15 16 17 18 19 20 21 -	int prim prim for	<pre>isPrime; es[0] = 2; es[1] = 3; eIndex = 2; ( p = 5; p &lt; isPrime = 1; for ( i : //if if ( i ; } if ( isP: prime ++pr: }</pre>	<pre>= N; p = p + 2 //assume that = 1; i &lt; prime p is divisibl p % primes[i] isPrime = 0; rime == 1 ) { es[primeIndex] imeIndex;</pre>	<pre>) {//iterate over a it is prime Index; i++ ){ e by some prime i, t == 0 ) = p;</pre>
≅Compiler ●Resources ▲Compile Log ✓D	ebug 🖻 Fin	d Results	Close		
✓Debug Add watch	Nex	t line	Continue	Next instruction	Send command to GDB: next *

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)

FrojectPrin	les - [Proj	ectPrimes.devj -	Debugg	jingj - Dev-C++	5.4.2 - 0
		nom ⊟eih			
Import Classes 0=000       Import per sector       Im	7 6 9 10 11 12 13 14 15 16 17 18 19 20 21	<pre>int isPrime primes[0] = primes[1] = primeIndex for ( p = 5 5     isPrime     for         for</pre>	; 2; 3; = 2; ; p <= ) = 1;//(i = //if p if ( p is ( isPrimes ++primes	N; p = p + 2 assume that 1; i < prime is divisibl * primes[i] Prime = 0; me == 1 ) { [primeIndex] eIndex;	<pre>) {//iterate over a it is prime Index; i++ ){ e by some prime i, t == 0 ) = p;</pre>
<sup>®</sup> Compiler <sup>●</sup> Resources <sup>@</sup> Compile Log <sup>✔</sup> De	ebug B Find	Results Close			
- Debug Add watch	<u>N</u> ext li	ine <u>C</u> ont	inue	Next instruction	Send command to GDB:next
Line: 15 Col: 1 !Insert D	one 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)

Project	Primes - [Proj	ectPrimes.dev] - [Deb	ougging] - Dev-C++	5.4.2 – 🔿 🗙
File Edit Search View Project Execute	Tools <u>C</u> VS <u>Wind</u>	low Help		
Primes = (2, 3, 5, 7, 1, 0, 4200297, 0,       #p = 9       # = 1       #primeIndex = 4       #isPrime = 0	0,0) 7 8 9 10 11¢ 12 14 14 15 16 17 18¢ 19 20 21	<pre>int isPrime; primes[0] = 2; primes[1] = 3; primeIndex = 2; for ( p = 5; p</pre>	<= N; p = p + 2 ;//assume that = 1; i < prime f p is divisibl ( p % primes[i] isPrime = 0; Prime == 1 ) { mes[primeIndex] rimeIndex;	<pre>2 ) {//iterate over a it is prime Index; i++ ) { le by some prime i, t == 0 ) = p;</pre>
Compiler Resources Compile Log	✓Debug <sup>©</sup> Find	Results Close		
Debug Add watch	<u>N</u> ext li	ne <u>C</u> ontinue	Next instruction	Send command to GDB:next *
Line: 13 Col: 1 !Inse	ert 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)



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

(Refer Slide Time: 07:55)

Project Classes Debug	primes.c							
B primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0}	7	int	isPrime;					
≊p = 9	-8	prim	es[0] = 2;					
15 i = 4	9	prime	es[1] = 3;					
sprimeIndex = 4	10	prime	eIndex = 2;		a second s			
sisPrime = 0	110	110 for ( p = 5; p <= N; p = p + 2 ) {//iterate over a						
+	12	12 isPrime = 1;//assume that it is prime						
	130	130 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;							
	10		}	1 1 1				
	10		11 ( 1SP	rime == 1 ) {				
	19		prim	imeIndex;	= p;			
	20		ttpr.	Imeridex;				
	21		1					
					>			
Compiler Resources Compile Log VDe	bug 🖻 Find	d Results	Close					
	•• •	P		No. of the other	Send command to GDB: next			

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)

Eile Edit Search View Project Execute Tool	s <u>C</u> VS <u>W</u> ind	ow <u>H</u> elp		
(goode) · · · · · · · · · · · · · · · · · · ·				
Project Classes Debug	primes.c			
<pre>■ primes = {2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0} ■ p = 11 = - 4</pre>	9 10	<pre>primes[1] = 3; primeIndex = 2; for ( n = 5; n </pre>		N (//itemte even e
l⊈i = 4	1.0	isDrime = 1:	-N; p = p + 2	it is primo
aisPrime = 0	130 14 15 16 17- 180 19 20 21- 22- 230	<pre>for ( i =</pre>	<pre>= 1; i &lt; prime p is divisibl p % primes[i] isPrime = 0; rime == 1) { es[primeIndex] imeIndex;</pre>	<pre>Index; i++ ) {     e by some prime i, t     == 0 )     = p; ++ ) {</pre>
				>
Compiler Resources Compile Log VDe	bug 🖻 Find	tesults Close		
Debug Add watch	<u>N</u> ext l	ne <u>C</u> ontinue	Next instruction	Send command to GDB: next

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)

speaker	-				
sprimes = (2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0) sp = 11 si = 0 sprimeIndex = 4 sisPrime = 0	10 1 11e 1 12 13e 1 14 15 16 17 - 18e 1 19 20 21 - 22 - 23e 1 24	For ( i For ( i For ( i pri	<pre>dex = 2; = 5; p &lt;= rime = 1;/ for ( i =</pre>	<pre>= N; p = p + 2 //assume that = 1; i &lt; prime p is divisibl p % primes[i] isPrime = 0; cime == 1 ) { s[primeIndex; primeIndex; i ', primes[i]);</pre>	<pre>) {//iterate over a it is prime Index; i++ ){ e by some prime i, t == 0 ) = p; ++ ){</pre>
<sup>®</sup> Compiler <sup>●</sup> Resources <sup>@</sup> Compile Log ✓De	bug B Find Re	sults <sup>®</sup> Clo	se		
≪Debug Add watch	Next line		Continue	Next instruction	Send command to GDB:next

We can check and print prime numbers.

(Refer Slide Time: 08:37)



So, print 2.

# (Refer Slide Time: 08:39)

Project Classes Debug	primes.c				
ℙ = (2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0) &p = 11 &i = 2 &primeIndex = 4 &sisPrime = 0	10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 <	<pre>primeI: for ( ) is for ( : pr:</pre>	<pre>ndex = 2; p = 5; p &lt;= Prime = 1;/ for ( i =</pre>	<pre>s N; p = p + 2 //assume that = 1; i &lt; prime p is divisibl p % primes[i] .sPrime = 0; time == 1) { ss[primeIndex] meIndex; i primeIndex; i , primes[i]);</pre>	<pre>) {//iterate over a it is prime Index; i++ ){ e by some prime i, t == 0 ) = p; ++ ){</pre>
Compiler Resources Compile Log VDe	bug B Find	Results Clo	ose		
Debug Add watch	Novtli	ino	Continuo	Novtinstruction	Send command to GDB: next

# Print 3

(Refer Slide Time: 08:40)



Print 5.

# (Refer Slide Time: 08:41)

ProjectPrim	nes - [Projec	tPrimes.dev] - [Deb	ugging] - Dev-C++	5.4.2 - 0			
Project Primes -           File Edit Search View Project Execute Tools CVS           Project Classes Debug         prim           ************************************		<pre>ProjectPrimes.devj - [Debugging] - Dev-C++ 5.4.2 - 6 * Window Help s.t primeIndex = 2; for ( p = 5; p &lt;= N; p = p + 2 ) {//iterate over a isPrime = 1;//assume that it is prime for ( i = 1; i &lt; primeIndex; i++ )(</pre>					
<sup>™</sup> Compiler *Resources <sup>4</sup> Compile Log <sup>√</sup> De	20 21 22 23 24 24 24 24	++p: } for ( 1 = 0; i < printf ("%d sults @Close	<pre>rimeIndex; &lt; primeIndex: i ", primes[i]); </pre>	++ ) {			
- Debug Add watch	Next line	Continue	Next instruction	Send command to God.			
Line: 23 Col: 1 Sinsert D	one parsing in	0.031 seconds					

# Print 7.

(Refer Slide Time: 08:42)



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

# (Refer Slide Time: 08:50)

	ProjectPri	mes - (ProjectPrim	es.dev] - [Debu	gging] - Dev-C++	5.42 - 8
He uu Proj sp	Veb Studio\Desl	dop) =	· //if	p is divisibl	e by some prime i, t.•
8p 81 80 80		25 } 26 print 27 retu	<pre>if (     isp     prim     ++pr     *     = 0; i &lt;     tf ("%d     tf ("%d     cf ("\n");     n 0;</pre>	<pre>p % primes[i] isPrime = 0; rime == 1 ) ( es[primeIndex] imeIndex; primeIndex; i ", primes[i]);</pre>	== 0 ) = p; ++ )(
Compiler * Resource	es 4 Compile Log -1	Debug 4 Find Results 4	Close		
- Siebug	Add watch	Next line	Continue	Next instruction	Send command to GDB:next *
Line: 28 Col	1 dinsert	Done parsing in 0.031 s	seconds		

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

(Refer Slide Time: 08:59)

	Pro	ectPrimes -	- [ProjectPrimes.dev] -	Dev-C++ 5.4.2		ж.
File Edit Search View Project Execute Tool UUIII available for the second seco		s CVS Windo primes.c 14 15 16 17 18 19 20 21 22 23 24 24 25	CVS Window Help primesc 14 15 16 17 18 19 20 21 22 23 24 25 3 24 25 3 25 3 25 3 26 27 27 20 27 20 20 20 21 22 3 23 24 25 3 25 3 26 27 27 20 27 20 20 20 20 20 20 20 20 20 20			
© Compiler Nesour	ces 4 Compile Log 1 O	ebug = Find Re	s as Close			
✓Debug	Add watch	Next line	e Continue	Next instruction	Send command to GDB: next	
Sep Executor	Yew GU window	into functi	on Skp function	Into instruction	⇒⇒prompt	4 19
Evaluate				+		
Line: 28 Co	E 1 dinsert D	one 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)



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)



I am going to run one more debug now.

(Refer Slide Time: 09:34)

File Edit Search View Project Execute Tools	es - [ProjectPrimes.d s CVS Window Help	ev] - [Debugging] - Dev-C++	5.4.2 – <del>a</del> ×
Project Classes Debug	primes c		
<pre>a primes = {-1, -1, 53, 0, 1, 0, 4200297, 0, 0,</pre>	0         primes[0]           9         primes[1]           10         primeInc           11e         for (p           12         isP           13e         I           14         I           15         I           16         I           17.         I           18e         I           19         20           21-         22-	<pre>2] = 2; a) = 3; lex = 2; = 5; p &lt;= N; p = p + 2; for ( i = 1; i &lt; prime)</pre>	<pre>&gt; {//iterate over a it is prime Index; i++ ){ e by some prime i, t. == 0 ) = p;</pre>
< >	tor (1	= II' 1 < nrimeInder: 14	++ 11
ECompiler Resources Compile Log Debug Add watch	bug <sup>®</sup> Find Results €Clos Next line	e Continue Next instruction	Send command to GDB; displa v

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)

ProjectPrim	es - [ProjectPrimes.	dev] - [Debuggi	ing] - Dev-C++	5.4.2 -	5 ×
File Edit Search View Project Execute Tools	CVS Window Help				
l≞ primes = (2, -1, 53, 0, 1, 0, 4200297, 0, 0, 0 l≞ p = 0 l≋ i = 1 l≊ primeIndex = 0 l≋ isPrime = 4231168	6 primes[ 9 primes] 10 primeIn 11 for ( p 12 13 14 15 16 17 18 19 20 21 22 } 23 for ( in 18 19 20 21 23 for ( in 23 23 for ( in 23 23 for ( in 23 14 15 16 17 18 18 19 19 10 10 10 10 10 10 10 10 10 10	<pre>D] = 2; 1] = 3; dex = 2; = 5; p &lt;= N rime = 1;//2 for (i = 1</pre>	<pre>N; p = p + 2 assume that : is divisible % primes[i] Prime = 0; ne == 1 ) { [primeIndex] eIndex;</pre>	<pre>) {//iterate c it is prime Index; i++ ){ e by some prime == 0 ) = p; ++ ){</pre>	vver a P i, t
۲ ک					>
Scompiler Resources Compile Log V De	bug Find Results Clos	ie			
Debug Add watch	Next line	Continue	Next instruction	Send command to GE	<sub>B:</sub> next ×
Line: 9 Col: 1 SInsert Do	ne parsing in 0.031 seco	nds			

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

### (Refer Slide Time: 09:48)

File Edit Search View Project Execute Too	s CVS Window Help
Project Classes Debug	primes.c
<pre>% primes = (2, 3, 53, 0, 1, 0, 4200297, 0, 0, 0 % p = 5 % i = 1 % primeIndex = 2 % isPrime = 4231168</pre>	<pre>6 primes[0] = 2; 9 primes[1] = 3; 10 primeIndex = 2; 11 for ( p = 5; p &lt;= N; p = p + 2 ) {//iterate over a 12 isPrime = 1;//assume that it is prime 13 for ( i = 1; i &lt; 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 ) { 10 primeS[i] == 0 } 10 primeS[i] == 1 } 10 primeS[i] == 0 } 10 primeS[i] == 0</pre>
< >> ≅Compiler % Resources	<pre>15</pre>
	Send command to GDB CONTIL

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)



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

(Refer Slide Time: 10:19)

ProjectPrin	nes - [ProjectPrimes.dev] - [Debugging] - Dev-C++ 5.4.2 - 🔹				
Elle Edit Search View Project Exerute Ioo Disease - Es au sea trans	is CVS Window Help				
Project Classes Debug	primes.c				
#primes = (2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0) #p = 9 #i = 3 #primeIndex = 4 #isPrime = 1	<pre>6  primes[0] = 2; 9  primes[1] = 3; 10  primeIndex = 2; 10  for ( p = 5; p &lt;= N; p = p + 2 ) {//iterate over a 10  isPrime = 1;//assume that it is prime 13  for ( i = 1; i &lt; primeIndex; i++ )( 14  //if p is divisible by some prime 1, t 15  if ( p % primes[i] == 0 ) 16  isPrime = 0; 17  ) 18  if ( isPrime == 1 ) { 19  primes[primeIndex] = p; 19  ++primeIndex; 21  } 22  } 23  for ( i = 0; i &lt; primeIndex; i++ ){</pre>				
Compiler * Resources # Compile Log * De	rbug II Find Results . Close				
Debug Add watch	Next line Continue Next instruction Send command to GDB-conti =				
Line: 12 Col. 1 :Insert D	one parsing in 0.031 seconds				

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)

File Edit Search View Project Classes Deb Seprimes = (2, 3, 5, Sep = 9 Sei = 3 SeprimeIndex = 4 Seprime = 1	Proj w Project Execute Iool ug ug 7, 1, 0, 4200297, 0, 0, 0}	primes.c 1 #include 2 //Program 3 #define 1 4 int main 5 { 6 []int 1 7 int : 8 prime 9 prime 10 prim 10 prime 10 prime 10 prime 10 prime 10 prime 10 p	<pre><stdio.h></stdio.h></pre>	<pre>Dev-C++ 5.4.2 P prime number N], primeInde N] primeInde N; p = p + 2 //assume that 1; i &lt; prime</pre>	<pre>- 5 s x; } {//iterate over it is prime Index; i++ ){</pre>	× ×
ECompiler Resour	ces 🛎 Compile Log 🗸 De	ebug 🗳 Find Results 🎙	Close			
✓Debug	Add watch	<u>N</u> ext line	Continue	Next instruction	Send command to GDB: CO	ntii *
Stop Execution	⊻iew CPU window	Into function	Skip function	Into instruction	->->prompt	~
Line: 27 Co	ol: 1 :Insert D	one parsing in 0.031	seconds			

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

### (Refer Slide Time: 10:43)

File Edit Search View Project Execute Iool		s CVS Wind	low <u>H</u> elp				
Project Classes Debug a primes = (2, 3, 5, 7, 1, 0, 4200297, 0, 0, 0) a p = 9 a i = 3 a primeIndex = 4 a isPrime = 1		15 16 17 - 18 = 19 20 21 - 22 - 23 = 24 25 - 26 27 27 27 27 27 27 27 27 27 27	<pre>} for ( i pri } printf return</pre>	<pre>if (</pre>	<pre>p % primes[i] sPrime = 0; ime == 1 ) { s[primeIndex] meIndex; primeIndex; i , primes[i]);</pre>	== 0 ) = p; ++ ){	,
Compiler Resources	●Compile Log ✔De	bug 🗳 Find	Results Clo	se			
✓ Debug	✓ Debug Add watch		ne	Continue	Next instruction	Send command t	o GDB: contii
	r - cout : 1	1.5.7		Chine from string of	late instruction	->->prompt	

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.