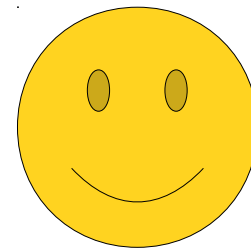
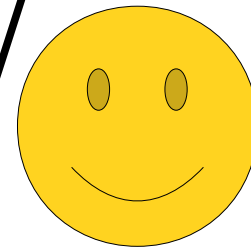


Assignment 0: Using the Debugger

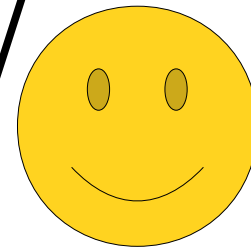
Hi everybody!



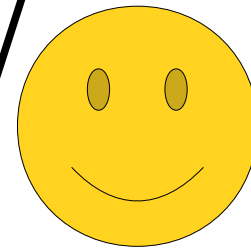
As part of Assignment 0, we'd like you to get a little bit of practice using the debugger in Qt Creator.



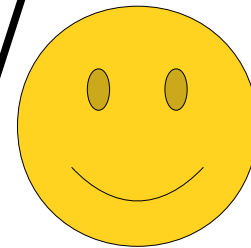
The debugger is a tool you can use to help see what your program is doing as you run it.



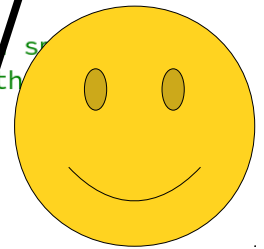
It's really useful for helping find errors in your programs, and the more practice you get with it, the easier it'll be to correct mistakes in the programs you write.



Think of this guide as a little tutorial walkthrough to help give you a sense of how to use the debugger and how to make sense of what you're seeing.



To start things off, open up the Name Hash program you ran in Part One of this assignment. Scroll down to the nameHash function so that you can see the entire function in your window.



```
42  * For t
43  * treat
44  * It th
45  * F_p, wh
46  * some smaller prime number q (you didn't expect a
47  * but we thought it might be fun!)
48  */
49  int nameHash(string first, string last){
50  /* This hashing scheme needs two prime numbers, a large prime
51  * prime. These numbers were chosen because their product is less than
52  * 2^31 - kLargePrime - 1.
53  */
54  static const int kLargePrime = 16908799;
55  static const int kSmallPrime = 127;
56
57  int hashVal = 0;
58
59  /* Iterate across all the characters in the first name, then the last
60  * name, updating the hash at each step.
61  */
62  for (char ch: first + last) {
63  /* Convert the input character to lower case. The numeric values of
64  * lower-case letters are always less than 127.
65  */
66  ch = tolower(ch);
67  hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68  }
69  return hashVal;
70  }
71
```

Activities Qt Creator Jan 4 3:09 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

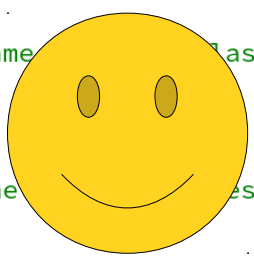
42 * For those of you who are more mathematically inclined, this function
43 * treats each character in the input name as a number between 0 and 128.
44 * It then uses them as coefficients in a polynomial over the finite field
45 * F_p , where p is a large prime number, and evaluates that polynomial at $x = \text{first} + \text{last}$ for CS106B,

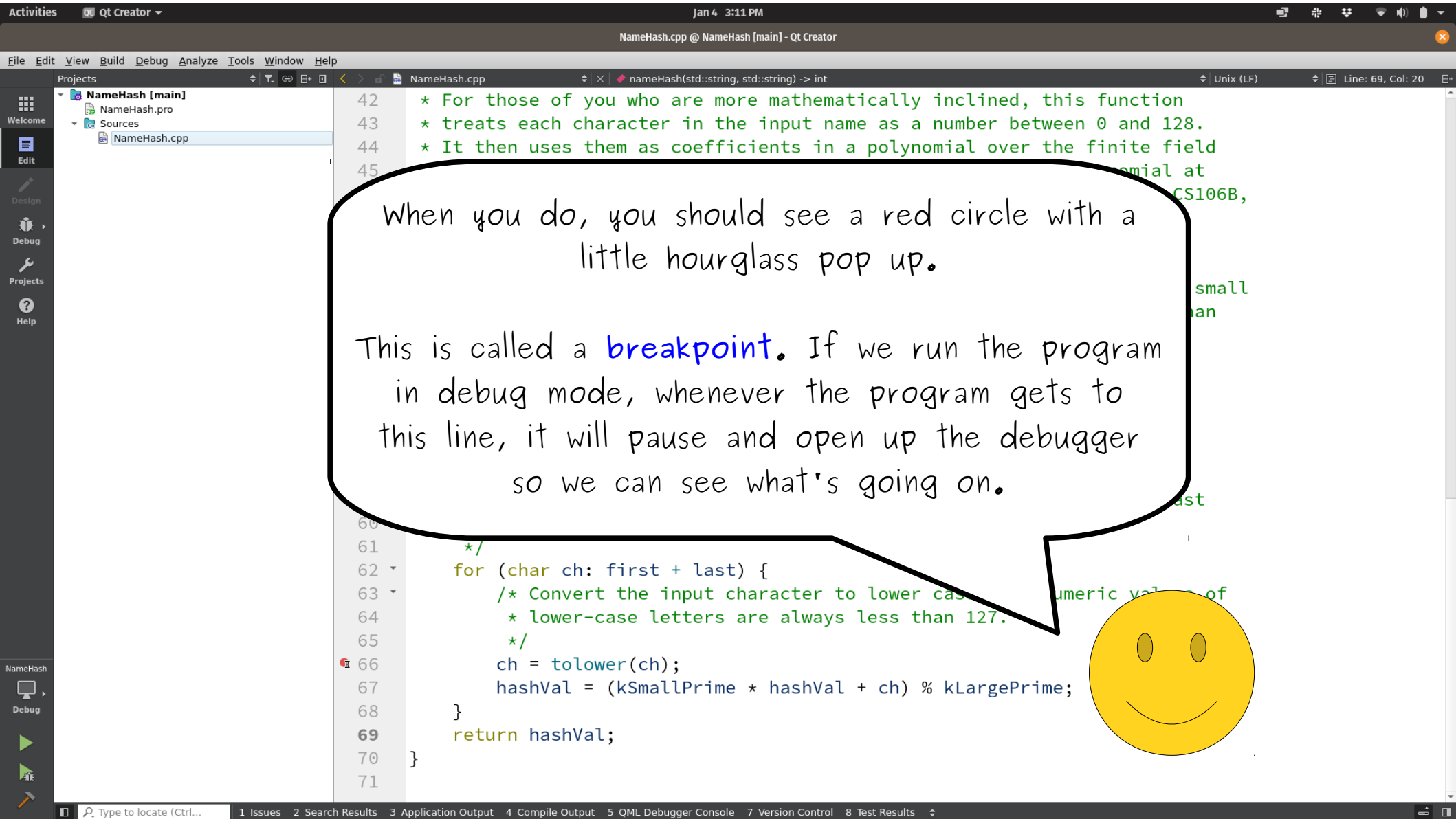
Move your mouse cursor so that it's in the space right before the line number for line 66.

Now, click the mouse!

56
57 `int hashVal = 0;`
58
59 `/* Iterate across all the characters in the input name from first to last`
60 `* name, updating the hash at each step.`
61 `*/`
62 `for (char ch: first + last) {`
63 `/* Convert the input character to lower case. The value of`
64 `* lower-case letters are always less than 127.`
65 `*/`
66 `ch = tolower(ch);`
67 `hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;`
68 `}`
69 `return hashVal;`
70 `}`
71

and a small less than





Activities Qt Creator

Jan 4 3:11 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

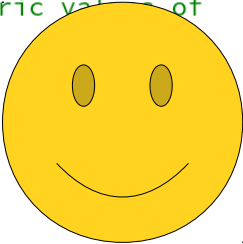
NameHash.pro

Sources

NameHash.cpp

```
42 * For those of you who are more mathematically inclined, this function
43 * treats each character in the input name as a number between 0 and 128.
44 * It then uses them as coefficients in a polynomial over the finite field
45 *  $F_p$ , where  $p$  is a large prime number, and evaluates that polynomial at
46 * some smaller prime number  $q$ . (You aren't expected to know this for CS106B,
47 * but we thought it might be fun!)
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers, a large prime and a small
51     * prime number  $q$ . (You aren't expected to know this for CS106B,
52     * but we thought it might be fun!)
53     */
54     for (char ch: first + last) {
55         /* Convert the input character to lower case. The numeric value of
56         * lower-case letters are always less than 127.
57         */
58         ch = tolower(ch);
59         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
60     }
61     return hashVal;
62 }
```

Now, we're going to run this program in debug mode. To do so, click on the "run in debug mode" button in the bottom-right corner of the screen. It's the one just below the regular green "run" button. When you do...



Qt Creator interface showing the NameHash.cpp file. The code implements a polynomial hashing function. A speech bubble points to the 'run in debug mode' button (a green bug icon) in the bottom-left toolbar. A yellow smiley face is also present.

... you should see something like this! Notice that a bunch of extra panels popped up in Qt Creator. We'll talk about what each of these windows mean in a second.

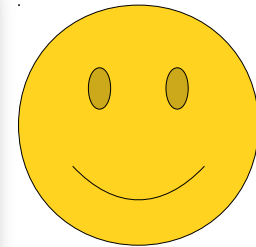
The screenshot shows the Qt Creator IDE interface for a project named 'NameHash'. The left sidebar contains a 'Projects' panel showing the project structure: 'NameHash [main]' containing 'NameHash.pro' and 'Sources' containing 'NameHash.cpp'. The main editor displays a C++ file with the following code:

```
45
46  * some smaller prime number q. (You aren
47  * but we thought it might be fun!)
48  */
49
50
51
52
53  What is your first name? |
54
55
56
57
58
59
60
61
62
63
64
65
66  cn = tolower(ch);
67  hashVal = (kSmallPrime * hashVal + ch) % kl
68  }
69  return hashVal;
```

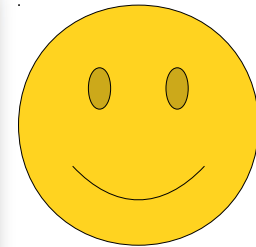
A 'NameHash Console' window is open, displaying the prompt 'What is your first name? |'. Below the console, a 'Debugger' window is open, showing the GDB for 'NameHash' with the application started. The debugger window has a table with the following data:

Level	Function	File	Line	Address	Number	Funct File	Line	Address	Condition	Ignore	Threads
1		...g) ...eHash.cpp	66	...5555b6782							(all)

The bottom status bar shows a search bar and a list of tabs: '1 Issues', '2 Search Results', '3 Application Output', '4 Compile Output', '5 QML Debugger Console', '7 Version Control', '8 Test Results'.



In the meantime, type in the first name **Ada** and hit enter, as shown here. We specifically want you to enter **Ada** here, *not your actual first name*.
(Unless your first name is Ada.)



Activities NameHash

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

45
46 * some smaller prime number q. (You aren't
47 * but we thought it might be fun!)
48 */
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66 `ch = tolower(ch);`
67 `hashVal = (kSmallPrime * hashVal + ch) % kL`
68 `}`
69 `return hashVal;`

NameHash Console

File Edit Options Help

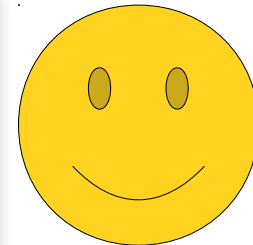
What is your first name? **Ada**
What is your last name? |

Debugger GDB for "NameHash" Threads: #12 Application started.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
					1eHash.cpp	66	...5555b6782			(all)

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Now, type in **Lovelace** as a last name, but
don't hit enter yet!



```
File Edit Options Help
What is your first name? Ada
What is your last name? Lovelace
```

```
45
46 * some smaller prime number q. (You aren
47 * but we thought it might be fun!)
48 */
49
```

```
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kl
68 }
69 return hashVal;
```

Debugger GDB for "NameHash" Threads: #12 Application started.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
					1eHash.cpp	66	...5555b6782			(all)

As soon as you hit enter, a bunch of things are going to pop up in Qt Creator. Don't panic! It's normal.

The image shows the Qt Creator IDE interface. The main editor displays the `NameHash.cpp` file with the following code:

```
45  
46  * some smaller prime number q. (You aren  
47  * but we thought it might be fun!)  
48  */  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  cn = tolower(ch);  
67  hashVal = (kSmallPrime * hashVal + ch) % kl  
68  }  
69  return hashVal;
```

A console window titled "NameHash Console" is open, showing the program's output:

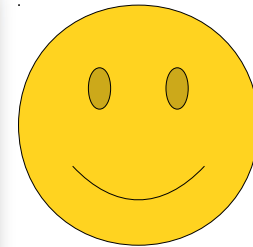
```
What is your first name? Ada  
What is your last name? Lovelace
```

A yellow smiley face is drawn next to the console window.

The debugger window at the bottom shows the application started. The table below represents the data shown in the debugger's "Threads" pane:

Level	Function	File	Line	Address	Number	Funct File	Line	Address	Condition	Ignore	Threads
1		...g	...eHash.cpp	66	66	...5555b6782			(all)

With that said, hit enter,
and watch the magic happen!



```
45  
46  * some smaller prime number q. (You aren  
47  * but we thought it might be fun!)  
48  */  
49
```

```
50  
51  
52 What is your first name? Ada  
53 What is your last name? Lovelace  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64
```

```
65  
66  ch = tolower(ch);  
67  hashVal = (kSmallPrime * hashVal + ch) % kl  
68  }  
69  return hashVal;
```

Debugger ▾ GDB for "NameHash" ▾ [Icons] Threads: #12 Application started.

Level	Function	File	Line	Address	Number	Funct File	Line	Address	Condition	Ignore	Threads
					1	...g) ...eHash.cpp	66	...5555b6782			(all)

Shazam! We're back in Qt Creator, and there's tons of values showing up everywhere.

Qt Creator interface showing the NameHash project. The main editor displays the NameHash.cpp file with the following code:

```
48
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers
51     * prime. These numbers were chosen because they are
52     *  $2^{31} - kLargePrime - 1$ .
53     */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first and last
60     * name, updating the hash at each step.
61     */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case. Note that
64         * lower-case letters are always less than upper-case
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

The right sidebar shows the Variable Explorer with the following variables:

Name	Value	Type
first	"AdaLovelace"	std::string
last	"A"	std::string
ch	"A"	char
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int

A yellow smiley face is drawn next to the Variable Explorer.

The bottom status bar shows the debugger is stopped at breakpoint 1 in thread 12. The stack trace is as follows:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEvent...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

The bottom status bar also shows the Build button and the text "Type to locate (Ctrl...)".

There's a lot going on right here. Let's see what's happening.

Qt Creator interface showing the NameHash project and the NameHash.cpp file.

Projects: NameHash [main], NameHash.pro, Sources, NameHash.cpp

Code (NameHash.cpp):

```
48
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers
51     * prime. These numbers were chosen because they are
52     *  $2^{31} - kLargePrime - 1$ .
53     */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first and last
60     * name, updating the hash at each step.
61     */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case. Note that
64         * lower-case letters are always less than upper-case
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger: GDB for "NameHash" (Thread: #12 NameHash, Stopped at breakpoint 1 in thread 12).

Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEventLoop()>::operator()			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Variables:

Name	Value	Type
first	"AdaLovelace"	std::string
last	"A"	std::string
ch	'A'	char
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int

Build: Build button visible.

Activities Qt Creator Jan 4 3:15 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - \text{kLargePrime} - 1$ .
```

First, notice that our red breakpoint now has a yellow arrow in it.

```
62 for (char ch: first + last)
63     /* Convert the input character to lower case
64      * lower-case letters are always
65      */
66     ch = tolower(ch);
67     hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71 }
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1	...	g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Build

Activities Qt Creator Jan 4 3:15 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers;
51     * prime. These numbers were chosen because the
52     *  $2^{31} - \text{kLargePrime} - 1$ .
```

This yellow arrow indicates where in the program we are right now. The program stopped running at this line because we hit that breakpoint you set earlier.

```
62 for (char ch: first + last)
63     /* Convert the input character to lower-case
64     * lower-case letters are always
65     */
66     ch = tolower(ch);
67     hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71 }
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start thread	pthread_create.c	463	0x7ffff6257590

Number	Funct	File	Line	Address	Condition	Ignore	Threads
1eHash.cpp	66	...5555b6782			(all)

Build

Activities Qt Creator Jan 4 3:15 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers;
51        * prime. These numbers were chosen because the
52        *  $2^{31} - \text{kLargePrime} - 1$ .
```

Whenever you pop up the debugger, it's good to figure out exactly where you are in the program that you're running, so you'll get into the habit of checking for this yellow arrow.

```
62 for (char ch: first + last)
63     /* Convert the input character to lower case
64        * lower-case letters are always
65        */
66     ch = tolower(ch);
67     hashVal = (kSmallPrime * hashVal + ch) % kPrime;
68 }
69 return hashVal;
70 }
71 }
```

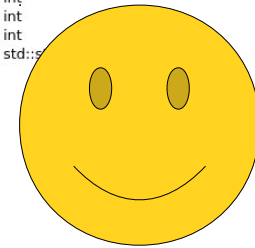
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start thread	pthread_create.c	463	0x7ffff6257590

Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	...	g) ...eHash.cpp	66	...5555b6782			(all)

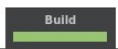
Build

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65 0x41	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::s

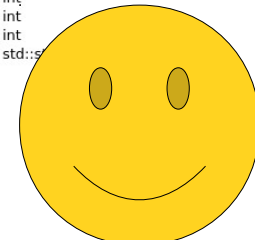


Next, let's take a look at this panel.
This is called the **call stack**.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								



Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::s

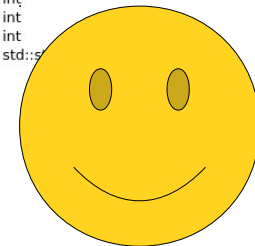


Right now, we know we're in the nameHash function, because our helpful friend the Yellow Arrow tells us exactly what line we're on!

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

```
48  */
49  int nameHash(string first, string last){
50  /* This hashing scheme needs two prime numbers:
51   * prime. These numbers were chosen because the
52   *  $2^{31} - kLargePrime - 1$ .
53   */
54   static const int kLargePrime = 16908799;
55   static const int kSmallPrime = 127;
56
57   int hashVal = 0;
58
59   /* Iterate across all the characters in the first
60    * name, updating the hash at each step.
61   */
62   for (char ch : first)
63   {
64       /*
65        *
66        */
67       hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68   }
69   return hashVal;
70 }
71
```

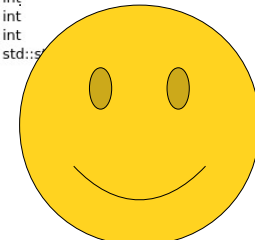
Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string



However, the yellow arrow can't tell us exactly how we got to this part of the program. What part of the program actually called nameHash?

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

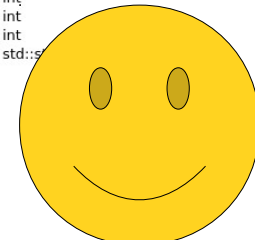
Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::s



The call stack can tell us exactly that!

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::s



Notice that the call stack lists a series of different functions in order. Here, it has nameHash (where we are now) at the top, and right below that is studentMain.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Activities Qt Creator Jan 4 3:15 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

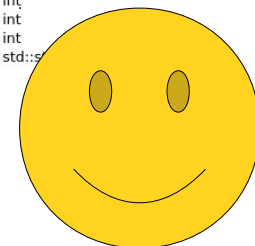
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	NameHash	NameHash.cpp	66	0x555555b6782	1g) ...eHash.cpp	66		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::call_once(handler_list(), QtGui::testBackendEnv::init())			0x5555555161b								
4	GThreadSt...			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start threa...	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Build



Go and double-click the call to studentMain on Level 2. When you do...

Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21 #include "simpio.h" // for getLine

22 using namespace std;

23 /* Prototype for the nameHash function. This lets u

24 * in main and then define it later in the program.

25 */

26 int nameHash(string first, string last);

27 int main() {

28 string first = getLine("What is your first name

29 string last = getLine("What is your last name?

30

31 int hashCode = nameHash(first, last);

32

33 cout << "The hash of your name is: " << hashCode

34 return 0;

35 }

36

37 /* This is the actual function that c

38 * to talk more about what hash functi

39 * the meantime, think of it as a functi

40 * of the input and produces a number

41 *

42 * For those of you who are more mathem

43 * treats each character in the input name as a num

Name


Name	Value	Type
first	"Ada"	std::string
hashCode	0	int
last	"Lovelace"	std::string

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1	...	g) ...eHash.cpp	66	...	555555b6782		(all)
2	studentMain	NameHash.cpp	31	0x5555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21

```
#include "simpio.h" // for getLine
using namespace std;

/* Prototype for the nameHash function. This lets u
 * in main and then define it later in the program.
 */
int nameHash(string first, string last);

int main() {
    string first = getLine("What is your first name
    string last = getLine("What is your last name?

    int hashCode = nameHash(first, last);

    cout << "The hash of your name is: " << hashCode
    return 0;
}

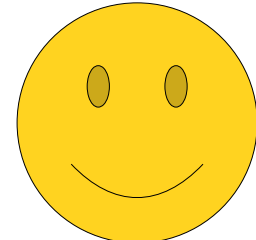
/* This is the actual function that c
 * to talk more about what hash funct
 * the meantime, think of it as a fun
 * of the input and produces a number
 *
 * For those of you who are more math
 * treats each character in the input
```

Debugger GDB for "NameHash" Threads: #12 Na

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x55...
2	studentMain	NameHash.cpp	31	0x55...
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x55...
4	GThreadStd::run()			0x55...
5	??			0x7ff...
6	start thread	pthread create.c	463	0x7ff...

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Con

Views Threads (all)



Notice that the yellow arrow points to Line 31. That line includes a call to the nameHash function. This is the part of the code that actually called nameHash, which is how we got to the line with the breakpoint!

Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21 #include "simpio.h" // for getLine

22 using namespace std;

23 /* Prototype for the nameHash function. This lets u

24 * in main and then define it later in the program.

25 */

26 int nameHash(string first, string last);

27 int main() {

28 string first = getLine("What is your first name

29 string last = getLine("What is your last name?

30

31 int hashValue = nameHash(first, last);

32

33 cout << "The hash of your name is: " << hashVal

34 return 0;

35 }

36

37 /* This is th

38 * to talk mo

39 * the meanti

40 * of the inp

41 *

42 * For those

43 * treats each ch

Name

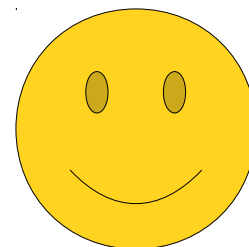
Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782								
2	studentMain	NameHash.cpp	31	0x555555b6595	1	...	eHash.cpp	66	...	5555b6782		(all)
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



Generally speaking, you can use the **call stack** as a way to see which function calls got us to the point where the program paused at the breakpoint!

Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21 #include "simpio.h" // for getLine

22 using namespace std;

23 /* Prototype for the nameHash function. This lets u

24 * in main and then define it later in the program.

25 */

26 int nameHash(string first, string last);

27 int main() {

28 string first = getLine("What is your first name

29 string last = getLine("What is your last name?

30

31 int hashCode = nameHash(first, last);

32

33 cout <<

34 return

35 }

36

37 /* This is

38 * to talk

39 * the meant

40 * of the input and produces a number.

41 *

42 * For those of you who are more mathematically inc

43 * treats each character in the input name as a num

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21 #include "simpio.h" // for getLine

22 using namespace std;

23 /* Prototype for the nameHash function. This lets u

24 * in main and then define it later in the program.

25 */

26 int nameHash(string first, string last);

27 int main() {

28 string first = getLine("What is your first name

29 string last = getLine("What is your last name?

30

31 int hashCode = nameHash(first, last);

32

33 cout <<

34 return

35 }

36

37 /* This is

38 * to talk

39 * the meant

40 * of the input and produces a number.

41 *

42 * For those of you who are more mathematically inc

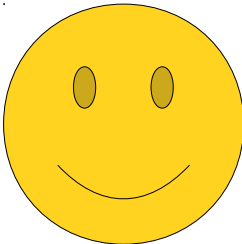
43 * treats each character in the input name as a num

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1	...	g) ...eHash.cpp	66	...	55555b6782		(all)
2	studentMain	NameHash.cpp	31	0x5555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



These grayed-out functions represent helper functions our libraries automatically call to help get your program set up.

Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21 #include "simpio.h" // for getLine

22 using namespace std;

23 /* Prototype for the nameHash function. This lets u

24 * in main and then define it later in the program.

25 */

26 int nameHash(string first, string last);

27 int main() {

28 string first = getLine("What is your first name

29 string last = getLine("What is your last name?

30

31 int hashCode = nameHash(first, last);

32

33 cout <<

34 return

35 }

36

37 /* This is

38 * to talk

39 * the meant

40 * of the input and produces a number.

41 *

42 * For those of you who are more mathematically inc

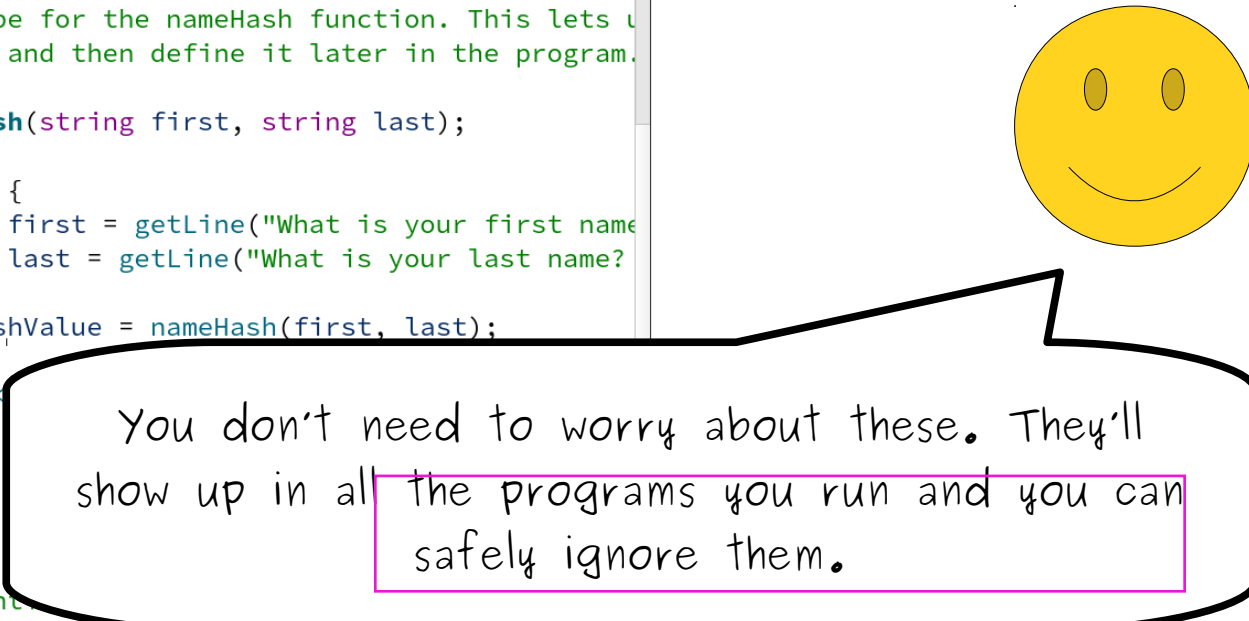
43 * treats each character in the input name as a num

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782								
2	studentMain	NameHash.cpp	31	0x555555b6595	1eHash.cpp	66	...5555b6782			(all)
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



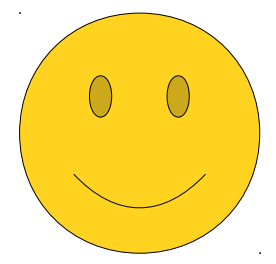
Projects

NameHash [main]
NameHash.pro
Sources
NameHash.cpp

Welcome
Edit
Design
Debug
Projects
Help

```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout <<
34     return
35 }
36
37 /* This is
38 * to talk
39 * the meant
40 * of the input and produces a number.
41 *
42 * For those of you who are more mathematically inc
43 * treats each character in the input name as a num
```

Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string



In the meantime, let's get back to our nameHash function. To do that, double-click on the nameHash entry at the top of the call stack. When you do...

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1g) ...eHash.cpp	66	...5555b6782			(all)
2	std::thread::id	std::thread.h	3	0x555555b6782								
3	std::thread::id	std::thread.h	3	0x555555b6161bc								
4	GThreadSt	GThreadSt		0x5555555f9476								
5	??	??		0x7ffff6143d84								
6	start thread	pthread create.c	463	0x7ffff6257590								

Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.


Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x5555555b6782
2	studentMain	NameHash.cpp	31	0x5555555b6595
3	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Name

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

You'll be teleported back here!



Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

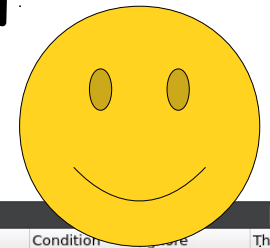
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters of the string 'last'
60      * and calculate the hash value.
61      */
62     for (int i = last.length() - 1; i >= 0; i--)
63     {
64         char ch = last[i];
65         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
66     }
67     return hashVal;
68 }
69
70
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::Function_handler<int (), QtGui::startBackgroundEvent...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Let's quickly recap what we've seen so far.



Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54
55
56
57
58
59
60
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

To set a breakpoint so that we can pause the program and look around, click in the margin just before the line number where you want to pause.

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

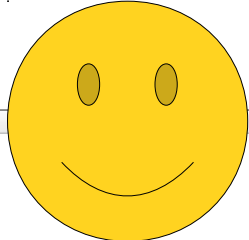
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54
55
56
57
58
59
60
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Once the breakpoint is reached, it will pull up all sorts of useful information.



Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54
55
56
57
58
59
60
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

The yellow arrow points out where we are right now.

Variable Inspector:

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65 0x41	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Debugger: GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Number	Func	File	Line	Address	Condition	Ignore	Threads
1g) ...eHash.cpp	66	...5555b6782			(all)

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54
55
56
57
58
59
60
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

The call stack shows us how we got into the current function.

for (char ch: first + last) {
 /* Convert the input character to lower case
 * lower-case letters are always less than
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), Qt...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	"A" 65 0x41	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

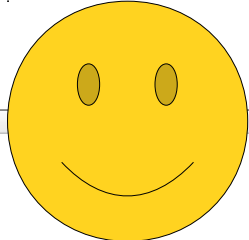
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51     * prime. These numbers were chosen because the
52     *  $2^{31} - kLargePrime - 1$ .
53     */
54
55
56
57
58
59
60
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64         * lower-case letters are always less than
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

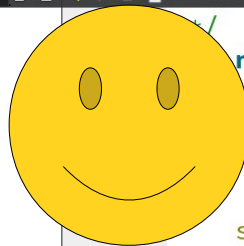
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Now, let's see how we can read the values of the variables in this function.



Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	"A" 65 0x41	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Look up at this panel over here.



Qt Creator interface showing the NameHash.cpp file and the Debug Console.

Code Editor (NameHash.cpp):

```
1  nameHash(string first, string last){
2      /* This hashing scheme needs two prime numbers:
3      * prime. These numbers were chosen because they
4      * are 2^31 - kLargePrime - 1.
5      */
6      static const int kLargePrime = 16908799;
7      static const int kSmallPrime = 127;
8
9      int hashVal = 0;
10
11     /* Iterate across all the characters in the first
12     * name, updating the hash at each step.
13     */
14     for (char ch: first + last) {
15         /* Convert the input character to lower case
16         * lower-case letters are always less than
17         */
18         ch = tolower(ch);
19         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
20     }
21     return hashVal;
22 }
```

Variable Inspector (Top Right):

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

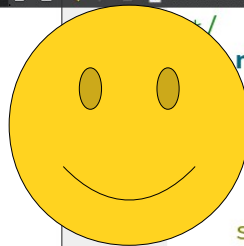
Debugger (Bottom):

Debugger: GDB for "NameHash" | Threads: #12 NameHash | Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Bottom Bar: 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

This window lets you take a look at all the values of the local variables that are in scope right now.



```
nameHash(string first, string last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because they
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65 0x41	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Depending on what OS you're using, these might be in a different order, and there might be some weird-looking ones in there in addition to nicer ones like `ch` and `hashVal`.

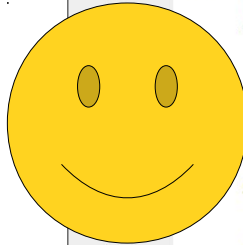


```
48  */
49  int nameHash(string first, string last){
50      /* This hashing scheme needs two prime numbers:
51       * prime. These numbers were chosen because they are
52       *  $2^{31} - kLargePrime - 1$ .
53       */
54      static const int kLargePrime = 16908799;
55      static const int kSmallPrime = 127;
56
57      int hashVal = 0;
58
59      /* Iterate across all the characters in the first string,
60       * name, updating the hash at each step.
61       */
62      for (char ch: first + last) {
63          /* Convert the input character to lower case.
64           * lower-case letters are always less than
65           */
66          ch = tolower(ch);
67          hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68      }
69      return hashVal;
70  }
71  }
```

Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

If we ignore the weird-looking ones, we can see some nice, familiar names.



```
48  */
49  int nameHash(const string& first, string last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

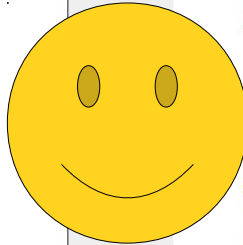
    int hashVal = 0;

    /* Iterate across all the characters in the first string,
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

```
type
std::string::iterator
std::string::iterator
std::string &&
char
std::string
int
int
int
std::string
Name Value Type
@0x7fff6058c80
"AdaLovelace"
'A' 65 0x41
"Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
```

Debugger ▾ GDB for "NameHash" ▾ [Icons] Threads: ➔ #12 NameHash ▾ Stopped at breakpoint 1 in thread 12.					Views ▾							
Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
➔ 1	nameHash	NameHash.cpp	66	0x555555b6782	● 1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

For example, here you can see the values of kLargePrime and kSmallPrime, which match the values they were declared with.



```
48  */
49  int nameHash(string first, string last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first name,
    * updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

```
__for_end @0x7ffff6058c80
__for_range "AdaLovelace"
ch 'A' 65 0x41
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
```

```
type
std::string::iterator
std::string::iterator
std::string &&
char
std::string
int
int
int
std::string
```

Name	Value	Type
------	-------	------

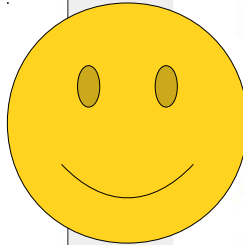
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

We can also see that, at this point, hashVal is still zero.



```
48  */
49  int nameHash(const string& first, string last){
    /* This hashing scheme needs two prime numbers;
    * prime. These numbers were chosen because they
    * are 2^31 - kLargePrime - 1.
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

```
__for_end @0x7ffff6058c80
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
```

```
type
std::string::iterator
std::string::iterator
std::string &&
char
std::string
int
int
int
std::string
```

Name	Value	Type
------	-------	------

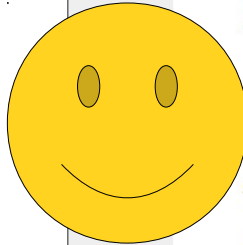
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

As we walk through the program one step at a time,
we'll see these values change.



```
48  */
49  int nameHash(const string& first, string last){
    /* This hashing scheme needs two prime numbers.
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string,
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than 128.
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

```
__for_end @0x7ffff6058c80
__for_range "AdaLovelace"
ch 'A' 65 0x41
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
```

```
type
std::string::iterator
std::string::iterator
std::string &&
char
std::string
int
int
int
std::string
```

Name	Value	Type
------	-------	------

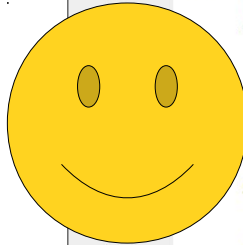
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555b6782	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x5555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Now, let's take a look at this for loop.



```
48  */
49  int nameHash(const string& first, string last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case
        * lower-case letters are always less than 128
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

```
__for_end @0x7ffff6058c80
__for_range "AdaLovelace"
ch 'A' 65 0x41
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
```

```
type
std::string::iterator
std::string::iterator
std::string &&
char
std::string
int
int
int
std::string
```

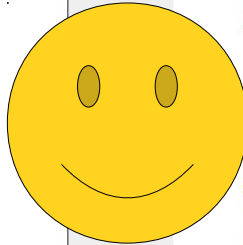
Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

This loop is a **range-based for loop**. It says
"for each character in the string first + last,
do something with that character."



```
48  */
49  int nameHash(const string& first, string& last){
    /* This hashing scheme needs two prime numbers.
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than 128.
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

```
__for_end @0x7ffff6058c80
__for_range "AdaLovelace"
ch 'A' 65 0x41
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
```

```
type
std::string::iterator
std::string::iterator
std::string &&
char
std::string
int
int
int
std::string
```

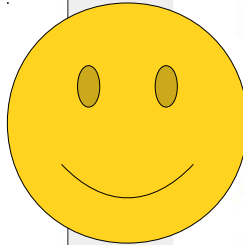
Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Remember (from a while back) that we entered
the name **Ada Lovelace**?



```
48  */
49  int nameHash(const string& first, string& last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

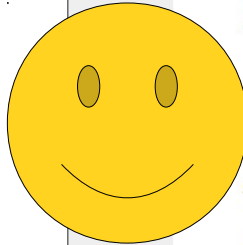
    int hashVal = 0;

    /* Iterate across all the characters in the first string,
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than 128.
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

Name	Value	Type
__for_end	@0x7ffff6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Debugger ▾ GDB for "NameHash" ▾					Threads: ➔ #12 NameHash		Stopped at breakpoint 1 in thread 12.					Views
Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
➔ 1	nameHash	NameHash.cpp	66	0x555555b6782	1	...	g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

If we take a look at the current value of the variable `ch`, we can see that it has the value `A`. That's the first letter of the name Ada Lovelace.



```
48  */
49  int nameHash(const string& first, string& last){
    /* This hashing scheme needs two prime numbers.
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than 128.
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

```
__for_end @0x7ffff6058c80
__for_range "Ada Lovelace"
ch 'A' 65 0x41
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
type
std::string::iterator
std::string::iterator
std::string &&
char
std::string
int
int
int
std::string
```

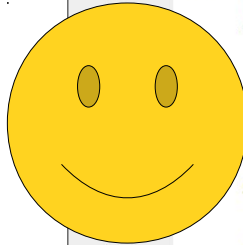
Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

So now we know where we are (line 66), how we got there (main called nameHash), and the values in the program at this point.



```
48  */
49  int nameHash(const string& first, string& last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

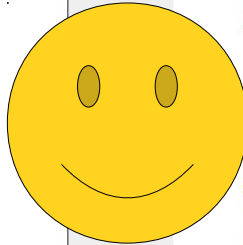
    int hashVal = 0;

    /* Iterate across all the characters in the first string
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case
        * lower-case letters are always less than 128
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

Name	Value	Type
__for_end	@0x7ffff6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Debugger ▾ GDB for "NameHash" ▾					Threads: ➔ #12 NameHash		Stopped at breakpoint 1 in thread 12.					Views ▾
Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
➔ 1	nameHash	NameHash.cpp	66	0x555555b6782	1	...	g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Now, let's do something really cool - we're going to run this program one line at a time, watching what happens at each step!



```
48  */
49  int nameHash(string first, string last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string,
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than 128.
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

Name	Value	Type
__for_end	@0x7ffff6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Debugger ▾ GDB for "NameHash" ▾					Threads: ➔ #12 NameHash		Stopped at breakpoint 1 in thread 12.					Views
Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
➔ 1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Activities Qt Creator

Jan 4 3:30 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

48 `*/`

49 `int nameHash(string first, string last){`

50 `/* This hashing scheme needs two prime numbers;`

51 `* prime. These numbers were chosen because the`

52 `* $2^{31} - kLargePrime - 1$.`

53 `*/`

54 `static const int kLargePrime = 16908799;`

55 `static const int kSmallPrime = 127;`

56

57 `int hashVal = 0;`

58

59 `/* Iterate across all the characters in the fir`

60 `* name, updating the hash at each step.`

61 `*/`

62 `for (ch`

63 `/*`

64 `*`

65 `*/`

66 `ch`

67 `has`

68 `}`

69 `return hashVal;`

70 `}`

71

Debugger GDB for "NameHash"


Threads: #12 NameHash

Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Right above the stack trace, you'll see there are some small button icons.



Activities Qt Creator

Jan 4 3:30 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers;

51 * prime. These numbers were chosen because the

52 * $2^{31} - kLargePrime - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (ch

63 /*

64 *

65 */

66 ch

67 has

68 }

69 return hashVal;

70 }

71 }

Name

Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

These buttons let you resume the program, stop the program, walk through it one line at a time, etc.

Debugger GDB for "NameHash"

Threads: #12 NameHash

Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1	...	g) ...eHash.cpp	66	...	55555b6782		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp


```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Move your mouse so that you're hovering over the button that's third from the left. If you hover over it, it should say "step over."



Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp


```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundThread::startBackgroundThread()>::operator()			0x555555b6161bc
4	GThreadStd::run()			0x555555b6f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Once you're confident that you're on the "Step Over" button - and not the "Step Into" or "Step Out" buttons - go and click it! When you do...



ActivitiesQt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

FileEditViewBuildDebugAnalyzeToolsWindowHelp

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers;

51 * prime. These numbers were chosen because the

52 * $2^{31} - kLargePrime - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (char ch: first + last) {

63 /* Convert the input character to lower cas

64 * lower-case letters are always less than

65 */

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % kL

68 }

69 return hashVal;

70 }

71 }

Name

Value

Type

for_begin @0x7fffc6058c78 std::string::iterator

for_end @0x7fffc6058c80 std::string::iterator

for_range "AdaLovelace" std::string &&

ch 'a' 97 0x61 char

first "Ada" std::string

hashVal 0 int

kLargePrime 16908799 int

kSmallPrime 127 int

last "Lovelace" std::string

DebuggerGDB for "NameHash"

LevelFunction

1nameHash

2studentMain

3std::_Function_handler<int (), G

4GThreadStd::run()

5??

6start thread

Type to locate (Ctrl...

1Issues2Search Results3Application Output4Compile Output5QML Debugger Console7Version Control8Test Results



...your window should look something like this.

Activities

Qt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

248

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

*/

int nameHash(string first, string last){

/* This hashing scheme needs two prime numbers.

* prime. These numbers were chosen because the

* 2^31 - kLargePrime - 1.

*/

static const int kLargePrime = 16908799;

static const int kSmallPrime = 127;

int hashVal = 0;

*/

Iterate across all the characters in the fir

* name, updating the hash at each step.

*/

for (char ch: first + last) {

/* Convert the input character to lower cas

* lower-case letters are always less than

*/

ch = tolower(ch);

hashVal = (kSmallPrime * hashVal + ch) % k

}

return hashVal;

}

Debugger

GDB for "NameHash"

Level Function

1 nameHash

2 studentMain

3 std::_Function_handler<int (), G

4 GThreadStd::run()

5 ??

6 start thread

Name

Value

Type

for_begin

@0x7fffc6058c78

std::string::iterator

for_end

@0x7fffc6058c80

std::string::iterator

for_range

"AdaLovelace"

std::string &&

ch

'a'

97

0x61

char

first

"Ada"

std::string

hashVal

0

int

kLargePrime

16908799

int

kSmallPrime

127

int

last

"Lovelace"

std::string

NameHash

Debug

Type to locate (Ctrl...

1 Issues

2 Search Results


3 Application Output

4 Compile Output

5 QML Debugger Console

7 Version Control

8 Test Results



Okay! A few things have changed. Let's see what's going on.

Activities

Qt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers;

51 * prime. These numbers were chosen because the

52 * $2^{31} - \text{kLargePrime} - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (char ch: first + last) {

63 /* Convert the input character to lower cas

64 * lower-case letters are always less than

65 */

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % k

68 }

69 return hashVal;

70 }

71 }

Name

Value

Type

for_begin @0x7ffc6058c78 std::string::iterator

for_end @0x7ffc6058c80 std::string::iterator

for_range "AdaLovelace" std::string &&

ch 'a' 97 0x61 char

first "Ada" std::string

hashVal 0 int

kLargePrime 16908799 int

kSmallPrime 127 int

last "Lovelace" std::string

Debugger

GDB for "NameHash"

Level Function

1 nameHash

2 studentMain

3 std::_Function_handler<int (), G

4 GThreadStd::run()

5 ??

6 start thread pthread create.c 463 0x7fff6257590

Type to locate (Ctrl...

1 Issues

2 Search Results

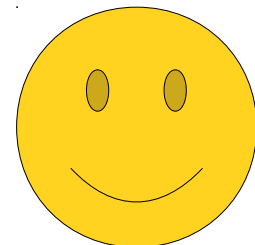
3 Application Output

4 Compile Output

5 QML Debugger Console

7 Version Control

8 Test Results



First, notice that our helpful Yellow Arrow friend is now pointing at line 67.

Activities

Qt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

*/

int nameHash(string first, string last){

/* This hashing scheme needs two prime numbers.

* prime. These numbers were chosen because the

* $2^{31} - \text{kLargePrime} - 1$.

*/

static const int kLargePrime = 16908799;

static const int kSmallPrime = 127;

int hashVal = 0;

*/

Iterate across all the characters in the first

* name, updating the hash at each step.

*/

for (char ch: first + last) {

/* Convert the input character to lower case

* lower-case letters are always less than

*/

ch = tolower(ch);

hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

}

return hashVal;

}

Name

Value

Type

__for_begin

@0x7ffc6058c78

std::string::iterator

__for_end

@0x7ffc6058c80

std::string::iterator

__for_range

"AdaLovelace"

std::string &&

ch

'a'

97

0x61

char

first

"Ada"

std::string

hashVal

0

int

kLargePrime

16908799

int

kSmallPrime

127

int

last

"Lovelace"

std::string

Debugger

GDB for "NameHash"

Level

Function

1

nameHash

2

studentMain

3

std::_Function_handler<int (), QThreadStd::run()

4

GThreadStd::run()

5

??

6

start thread

Type to locate (Ctrl...

1 Issues

2 Search Results


3 Application Output

4 Compile Output

5 QML Debugger Console

7 Version Control

8 Test Results



We're now at the line right after the one where we stopped. You just ran a single line of the program! Pretty cool!

Activities

Qt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48

*/

49

int nameHash(string first, string last){

50

/* This hashing scheme needs two prime numbers;

51

* prime. These numbers were chosen because the

52

* 2^31 - kLargePrime - 1.

53

*/

54

static const int kLargePrime = 16908799;

55

static const int kSmallPrime = 127;

56

57

int hashVal = 0;

58

59

/* Iterate across all the characters in the fir

60

* name, updating the hash at each step.

61

*/

62

for (char ch: first + last) {

63

/* Convert the input character to lower cas

64

* lower-case letters are always less than

65

*/

66

ch = tolower(ch);

67

hashVal = (kSmallPrime * hashVal + ch) % k

68

}

69

return hashVal;

70

}

71

Name

Value

Type

__for_begin

@0x7fffc6058c78

std::string::iterator

__for_end

@0x7fffc6058c80

std::string::iterator

__for_range

"AdaLovelace"

std::string &&

ch

'a'

97

0x61

char

first

"Ada"

std::string

hashVal

0

int

kLargePrime

16908799

int

kSmallPrime

127

int

last

"Lovelace"

std::string

Debugger

GDB for "NameHash"

Level

Function

1

nameHash

2

studentMain

3

std::_Function_handler<int (), Q

4

GThreadStd::run()

5

??

6

start thread

pthread create.c

463

0x7ffff6257590

Type to locate (Ctrl...

1 Issues

2 Search Results


3 Application Output

4 Compile Output

5 QML Debugger Console

7 Version Control

8 Test Results



so what did that line of code do?

Activities Qt Creator Jan 4 3:42 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - \text{kLargePrime} - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first and last
60      * name, updating the hash at each step.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case. Note that
64          * lower-case letters are always less than upper-case
65          * letters.
66          */
67         ch = tolower(ch);
68         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69     }
70     return hashVal;
71 }
```


Debugger GDB for "NameHash"

Level	Function
1	nameHash
2	studentMain
3	std::_Function_handler<int (), QThreadStd::run()
4	GThreadStd::run()
5	??
6	start thread

pthread create.c 463 0x7ffff6257590

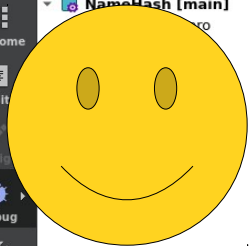
Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'a' 97	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Type to locate (Ctrl+Shift+F) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



This line converts ch to lower case. The tolower function takes in a character and returns a lower-case version of it, so this overwrites ch with a lower-case version of itself.

You can actually see this by looking at the values panel over on the side!



```
50 int nameHash(string first, string last){
51     /* This hashing scheme needs two prime numbers:
52      * prime. These numbers were chosen because they
53      *  $2^{31} - \text{kLargePrime} - 1$ .
54      */
55     static const int kLargePrime = 16908799;
56     static const int kSmallPrime = 127;
57
58     int hashVal = 0;
59
60     /* Iterate across all the characters in the first
61      * name, updating the hash at each step.
62      */
63     for (char ch: first + last) {
64         /* Convert the input character to lower case
65          * lower-case letters are always less than
66          */
67         ch = tolower(ch);
68         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69     }
70     return hashVal;
71 }
```

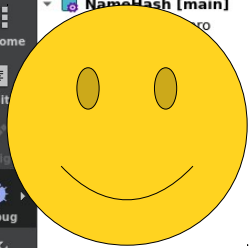
	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'a' 97	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1	...	g) ...eHash.cpp	66		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Notice that the value associated with `ch` has changed from 'A' to 'a' - it's now in lower-case!



```
int nameHash(string first, string last){
    /* This hashing scheme needs two prime numbers;
    * prime. These numbers were chosen because the
    *  $2^{31} - \text{kLargePrime} - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

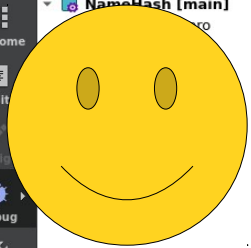
	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"Ada Lovelace"	std::string &&
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1	...	g) ...eHash.cpp	66		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

If you'll notice, this value is in red while all the other values are in black.



```
int nameHash(string first, string last){
    /* This hashing scheme needs two prime numbers;
    * prime. These numbers were chosen because the
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"Ada Lovelace"	std::string &&
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1	...	g) ...eHash.cpp	66		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

This indicates that the value here has changed since the previous step. This is a really useful way to keep track of what's changing as you run the program.



```
50 int nameHash(string first, string last){
51     /* This hashing scheme needs two prime numbers;
52     * prime. These numbers were chosen because the
53     *  $2^{31} - kLargePrime - 1$ .
54     */
55     static const int kLargePrime = 16908799;
56     static const int kSmallPrime = 127;
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60     * name, updating the hash at each step.
61     */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64         * lower-case letters are always less than
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71 }
```

	Value	Type
for_begin	@0x7ffc6058c78	std::string::iterator
for_end	@0x7ffc6058c80	std::string::iterator
for_range	"Ada Lovelace"	std::string &&
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Name	Value	Type
------	-------	------

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1	...	g) ...eHash.cpp	66		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Activities Qt Creator Jan 4 3:42 PM NameHash.cpp @ NameHash [main] - Qt Creator

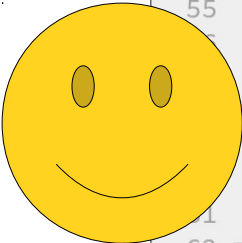
File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Welcome Edit Design Debug Projects Help

Now, let's take a look at line 67, where we are right now.



```
54
55
56 start = 127;
57
58 int hashVal = 0;
59
60 /* Iterate across all the characters in the first
61 * name, updating the hash at each step.
62 */
63 for (char ch: first + last) {
64     /* Convert the input character to lower case
65     * lower-case letters are always less than
66     */
67     ch = tolower(ch);
68     hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69 }
70 return hashVal;
71 }
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	67	0x555555b6790
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Number	Funct	File	Line	Address	Condition	Ignore	Threads
1g) ...eHash.cpp	66		(all)

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

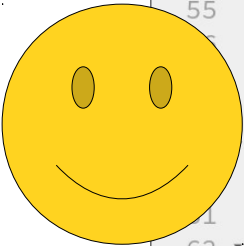
Activities Qt Creator Jan 4 3:42 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Not gonna lie, this is a pretty dense line of code. It performs some weird sort of mathematical calculation on a bunch of different values.



```
54
55
56 // ...
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63     /* Convert the input character to lower case
64     * lower-case letters are always less than
65     */
66     ch = tolower(ch);
67     hashVal = (kSmallPrime * hashVal + ch) % kSmallPrime;
68 }
69 return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funcnt	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1g) ...eHash.cpp	66		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:42 PM NameHash.cpp @ NameHash [main] - Qt Creator

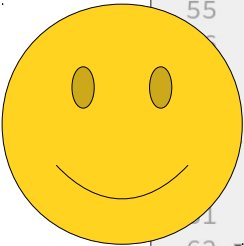
File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Welcome Edit Design Debug Projects Help

Fundamentally, though, it's just computing some weird function of some values and stashing it into hashVal.



```
54
55
56 start = 127;
57
58 int hashVal = 0;
59
60 /* Iterate across all the characters in the first
61 * name, updating the hash at each step.
62 */
63 for (char ch: first + last) {
64     /* Convert the input character to lower case
65     * lower-case letters are always less than
66     */
67     ch = tolower(ch);
68     hashVal = (kSmallPrime * hashVal + ch) % kMaxVal;
69 }
70 return hashVal;
71 }
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:42 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Let's go run that line of code and see what happens!



```
54
55
56 start = 127;
57
58 int hashVal = 0;
59
60 /* Iterate across all the characters in the first
61 * name, updating the hash at each step.
62 */
63 for (char ch: first + last) {
64     /* Convert the input character to lower case
65     * lower-case letters are always less than
66     */
67     ch = tolower(ch);
68     hashVal = (kSmallPrime * hashVal + ch) % kMaxVal;
69 }
70 return hashVal;
71 }
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1eHash.cpp	66	...			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:42 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp


```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers;
51        * prime. These numbers were chosen because the
52        *  $2^{31} - kLargePrime - 1$ .
53        */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60        * name, updating the hash at each step.
61
62
63
64
65
66
67     return hashVal;
68 }
69
70
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1	...	g) ...eHash.cpp	66	...	5555b6782		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startB...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread create.c	463	0x7ffff6257590								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Hover over the "Step Over" button, confirm that the button you're clicking really is "Step Over," and click it! When you do...



Activities

Qt Creator

Jan 4 3:48 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

*/

int nameHash(string first, string last){

/* This hashing scheme needs two prime numbers:

* prime. These numbers were chosen because the

* $2^{31} - kLargePrime - 1$.

*/

static const int kLargePrime = 16908799;

static const int kSmallPrime = 127;

int hashVal = 0;

*/

/* Iterate across all the characters in the fir

* name, updating the hash at each step.

*/

for (char ch: first + last) {

/* Convert the input character to lower cas

* lower-case letters are always less than

*/

ch = tolower(ch);

hashVal = (kSmallPrime * hashVal + ch) % kL

}

return hashVal;

}

Name

Value

Type

for_begin

@0x7fffc6058c78

std::string::iterator

for_end

@0x7fffc6058c80

std::string::iterator

for_range

"AdaLovelace"

std::string &&

ch

'a'

97

0x61

char

first

"Ada"

std::string

hashVal

97

int

kLargePrime

16908799

int

kSmallPrime

127

int

last

"Lovelace"

std::string

Debugger

GDB for "NameH

Level

Function

1

nameHash

2

studentMain

3

std::_Function_handler<int (), QtGui::startBackgroundDeve...

4

GThreadStd::run()

5

??

6

start thread

pthread create.c

463

0x7ffff6257590

Type to locate (Ctrl...

1 Issues

2 Search Results

3 Application Output


4 Compile Output

5 QML Debugger Console

7 Version Control

8 Test Results

... you'll end up with something like this!



Activities

Qt Creator

Jan 4 3:48 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers:

51 * prime. These numbers were chosen because the

52 * $2^{31} - kLargePrime - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (char ch: first + last) {

63 /* Convert the input character to lower cas

64 * lower-case letters are always less than

65 */

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % k

68 }

69 return hashVal;

70 }

71 }

Name

Value

Type

__for_begin @0x7fffc6058c78 std::string::iterator

__for_end @0x7fffc6058c80 std::string::iterator

__for_range "AdaLovelace" std::string &&

ch 'a' 97 0x61 char

first "Ada" std::string

hashVal 97 int

kLargePrime 16908799 int

kSmallPrime 127 int

last "Lovelace" std::string

Debugger

GDB for "NameH-

Level Function

1 nameHash

2 studentMain

3 std::_Function_handler<int (), QtGui::startBackgroundDeve...

4 GThreadStd::run()

5 ??

6 start thread pthread create.c 463 0x7ffff6257590

1 Issues

2 Search Results

3 Application Output

4 Compile Output

5 QML Debugger Console

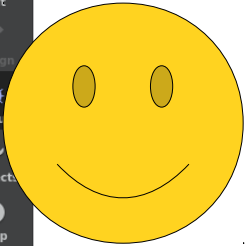
7 Version Control

8 Test Results



Let's see what's changed.

First, notice that the value stored in hashVal changed to 97. We know that it changed because the value is in red, and we know that nothing else changed because nothing else is in red!



```
51  /* This hashing scheme needs two prime numbers;
52  * prime. These numbers were chosen because the
53  *  $2^{31} - kLargePrime - 1$ .
54  */
55  static const int kLargePrime = 16908799;
56  static const int kSmallPrime = 127;
57
58  int hashVal = 0;
59
60  /* Iterate across all the characters in the first
61  * name, updating the hash at each step.
62  */
63  for (char ch: first + last) {
64      /* Convert the input character to lower case
65      * lower-case letters are always less than
66      */
67      ch = tolower(ch);
68      hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69  }
70  return hashVal;
71 }
```

Variable	Value	Type
first	@0x7fff6058c78	std::string::iterator
last	@0x7fff6058c80	std::string::iterator
ch	"AdaLovelace"	std::string &&
hashVal	97	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb	1	...	g) ...eHash.cpp	66	...	5555b6782		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Activities Qt Creator Jan 4 3:48 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

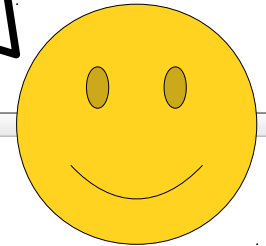
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51     * prime. These numbers were chosen because the
52     *  $2^{31} - kLargePrime$ 
53     */
54     static
55     static
56
57     int has
58
59     /* Iter
60     * name
61     */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64         * lower-case letters are always less than
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kL
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Second, notice that we're back up at the top of the for loop, since that's where the yellow arrow is pointing. We ended up back here because this is the next line that gets executed.



Activities Qt Creator Jan 4 3:48 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

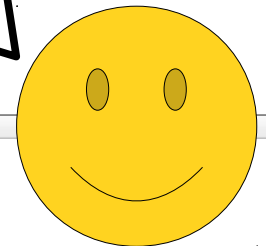
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters in the string.
60      * nameHash will calculate the hash value for the string.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than 128.
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEventLoop()>::operator()			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

We just single-stepped through a single iteration of that loop! Pretty cool!



Activities Qt Creator Jan 4 3:48 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

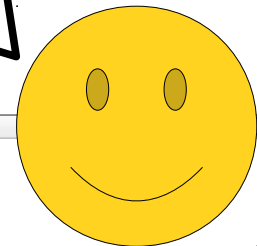
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters of the strings.
60      * nameHash iterates over the characters of the strings
61      * and calculates the hash value.
62     */
63     for (char ch: first + last) {
64         /* Convert the input character to lower case.
65          * lower-case letters are always less than
66          * upper-case letters.
67          */
68         ch = tolower(ch);
69         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
70     }
71     return hashVal;
72 }
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEventLoop()>::operator()			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Let's go do it again!



Activities Qt Creator Jan 4 3:48 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

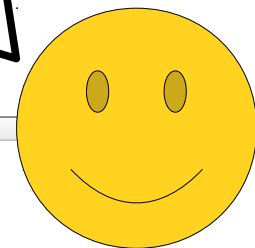
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters in the strings.
60      * nameHash will iterate over the characters in the strings.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than 128.
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundThread::<void*>::operator()() const			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Again, move your mouse over the Step Over button (and make sure it says "Step Over" and not something else!), then click it.



Activities Qt Creator Jan 4 3:50 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

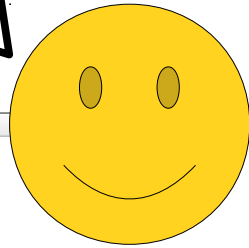
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over each character in the first and last names.
60      * nameHash will calculate the hash for each character.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than 128.
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEventLoop()			0x5555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Now we're here! Notice that ch now has the value 'd', which is the second letter of the name Ada.



Activities Qt Creator Jan 4 3:50 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

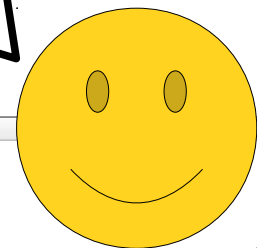
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters of the strings.
60      * nameHash will iterate over the characters of the strings.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than 128.
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundThreadEventLoop...			0x5555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Go click "Step Over" again to run this line of code.



Activities Qt Creator Jan 4 3:52 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

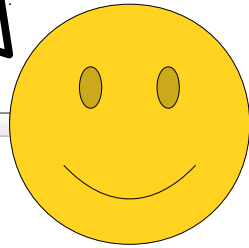
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters in the first and last strings.
60      * nameHash will convert the characters to lower case.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than 128.
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEvent...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl+K) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

You should be here now. Notice that none of the values changed. That makes sense, since all we did was convert a lower-case 'd' to a lower-case 'd'.



Activities Qt Creator Jan 4 3:52 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

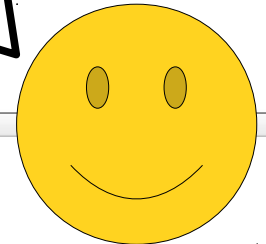
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters in the string
60      * nameHash.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	67	0x555555b6790
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEvent...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Now, click "Step Over" one more time.



Activities Qt Creator Jan 4 3:53 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers;
51     * prime. These numbers were chosen because the
52     *  $2^{31} - kLargePrime - 1$ .
53     */
54     static const int kLargePrime = 16908799;
55
56
57
58 }
59 return hashVal;
60
61 }
```


Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Name	Value	Type
__for_begin	@0x7ffff6058c78	std::string::iterator
__for_end	@0x7ffff6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	"d" 100 0x64	char
first	"Ada"	
hashVal	????	
kLargePrime	16908799	
kSmallPrime	127	
last	"Lovelace"	

Look here!

You'll now be at this point in the program. We've covered up the value of hashVal in this image, because at this point you should be able to see what hashVal is by reading the value in the side pane. This is the special value we want you to tell us when submitting the assignment!



Activities Qt Creator Jan 4 3:53 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51     * prime. These numbers were chosen because the
52     *  $2^{31} - kLargePrime - 1$ .
53     */
54
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64         * lower-case letters are always less than
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

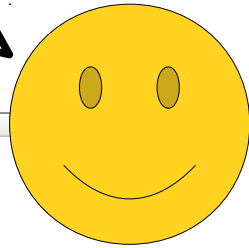
Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Name Value Type

__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'd' 100	char
first	"Ada"	std::string
hashVal	????	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

To finish up this section on the debugger, we'd like to show you two last little techniques that you might find useful when debugging programs.



Activities Qt Creator Jan 4 3:53 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51     * prime. These numbers were chosen because the
52     *  $2^{31} - kLargePrime - 1$ .
53     */
54
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64         * lower-case letters are always less than
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

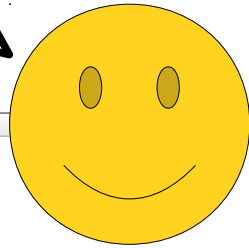
Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Name Value Type

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'd' 100	char
first	"Ada"	std::string
hashVal	????	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

To start this off, click on the the breakpoint that we set earlier in the program. If you do...



Activities Qt Creator Jan 4 3:57 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

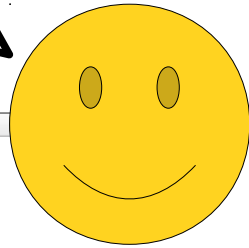
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - \text{kLargePrime} - 1$ .
53      */
54
55     for (char ch: first + last) {
56         /* Convert the input character to lower case
57          * lower-case letters are always less than
58          */
59         ch = tolower(ch);
60         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
61     }
62     return hashVal;
63 }
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

... it should clear the breakpoint. Now, if we were to run this program again in debug mode, it would not stop at this point, since nothing's telling it to!



Activities Qt Creator Jan 4 3:57 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

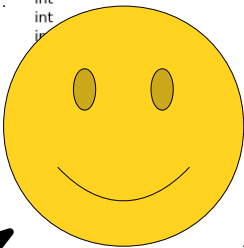
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb								
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Now, take a look back at these buttons.



Activities Qt Creator Jan 4 3:57 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - \text{kLargePrime} - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

Name Value Type

__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'd' 100	char
first	"Ada"	std::string
hashVal	????	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb								
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Hover your mouse over the one that's on the far right. When you hover over it, it should say "step out."

Activities Qt Creator

Jan 4 3:57 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

48 `*/`

49 `int nameHash(string first, string last){`

50 `/* This hashing scheme needs two prime numbers;`

51 `* prime. These numbers were chosen because the`

52 `* $2^{31} - kLargePrime - 1$.`

53 `*/`

54 `static const int kLargePrime = 16908799;`

55 `static const int kSmallPrime = 127;`

56

57 `int hashVal = 0;`

58

59 `/* Iterate across all the characters in the fir`

60 `* name, updating the hash at each step.`

61 `*/`

62 `for (ch`

63 `/*`

64 `*`

65 `*/`

66 `ch`

67 `has`

68 `}`

69 `return hashVal;`

70 `}`

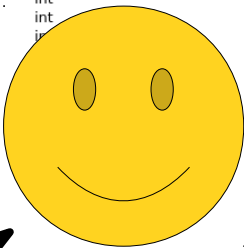
71

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb								
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Don't click just yet. But when you do click, it will run the rest of the nameHash function until it finishes and returns.



Activities Qt Creator

Jan 4 3:57 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

48 `*/`

49 `int nameHash(string first, string last){`

50 `/* This hashing scheme needs two prime numbers;`

51 `* prime. These numbers were chosen because the`

52 `* $2^{31} - kLargePrime - 1$.`

53 `*/`

54 `static const int kLargePrime = 16908799;`

55 `static const int kSmallPrime = 127;`

56

57 `int hashVal = 0;`

58

59 `/* Iterate across all the characters in the fir`

60 `* name, updating the hash at each step.`

61 `*/`

62 `for (ch`

63 `/*`

64 `*`

65 `*/`

66 `ch`

67 `has`

68 `}`

69 `return hashVal;`

70 `}`

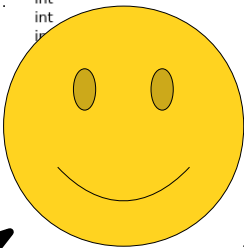
71

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb								
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::function_handler<int (), QtGui::startBackground			0x55555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Now, go click that button. If you did everything right...



Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

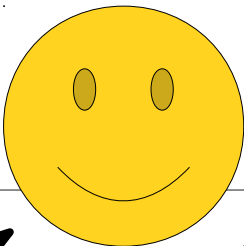
```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashVa
34     return 0;
35 }
36
37 /* This is
38 * to talk
39 * the mean
40 * of the i
41 *
42 * For thos
43 * treats e
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x555555b6595
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

... you should end up with something that looks like this!



Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

returned value 1967457 int

Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21

```
#include "simpio.h" // for getLine
using namespace std;

/* Prototype for the nameHash function. This lets u
 * in main and then define it later in the program.
 */
int nameHash(string first, string last);

int main() {
    string first = getLine("What is your first name
    string last = getLine("What is your last name?

    int hashValue = nameHash(first, last);

    cout << "The hash of your name is: " << hashVa
    return 0;
}

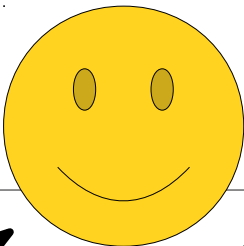
/* This is
 * to talk
 * the mean
 * of the i
 *
 * For thos
 * treats e
```

22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

Line: 31, Col: 5

Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

returned value 1967457 int



Let's take a minute to get our bearings.
Where exactly are we?

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	studentMain	NameHash.cpp	31	0x555555b6595								
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
3	GThreadStd::run()			0x5555555f9476								
4	??			0x7ffff6143d84								
5	start_thread	pthread_create.c	463	0x7ffff6257590								
6	clone	clone.S	95	0x7ffff5e30223								

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21

```
#include "simpio.h" // for getLine
using namespace std;

/* Prototype for the nameHash function. This lets u
 * in main and then define it later in the program.
 */
int nameHash(string first, string last);

int main() {
    string first = getLine("What is your first name
    string last = getLine("What is your last name?

    int hashValue = nameHash(first, last);

    cout << "The hash of your name is: " << hashVa
    return 0;
}

/* This is
 * to talk
 * the mean
 * of the i
 *
 * For thos
 * treats e
```

31

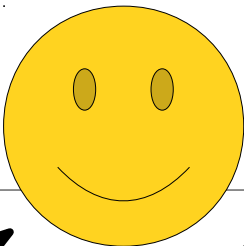
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x555555b6595
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Type to locate (Ctrl...

Well, the yellow arrow indicates that we're back in main again. Cool!



Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30     int hashValue = nameHash(first, last);
31
32     cout << "The hash of your name is: " << hashVal
33     return 0;
34 }
35
36
37 /* This is th
38 * to talk m
39 * the meant
40 * of the in
41 *
42 * For those
43 * treats ea
```

Debugger GDB for "NameHash"

Level Function

- 1 studentMain
- 2 std::Function_handler<int (), QtG
- 3 GThreadStd::run()
- 4 ??
- 5 start_thread
- 6 clone

Debugger Variables

Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

Debugger Console

returned value 1967457 int

We can see that the nameHash function returned 1967457. Thanks, debugger!

(A note: it seems like on some Macs, this number doesn't display. Don't worry if you don't see it - just continue on as usual.)

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashVal
34     return 0;
35 }
36
37 /* This is the actual function that computes the ha
38 * to talk more about what hash functions do later
39 the meantime, think of it as a function that scr
40 the input and produces a number.
```

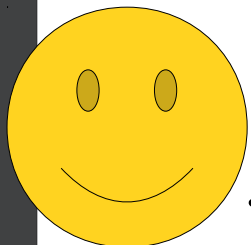
Name	Value	Type
hashValue	0	int

returned value 1967457 int

Name	Value	Type
------	-------	------

Views Line Address Condition Ignore Threads

Type to locate (Ctrl...



But if we look up over here, we see that hashValue isn't storing 1967457, even though that's what was returned.

(You might see a number other than 0 on your system - that's okay.)



Activities Qt Creator

Jan 4 4:02 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Welcome Edit Design Debug Projects Help

```
19 #include "simpio.h"
20 using namespace std;
21
22 /* Prototype for nameHash
23  * in main and
24  */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first;
29     string last = getLine("What is your last name? ");
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashValue;
34     return 0;
35 }
36
37 /* This is the actual function that computes the hash value.
38  * to talk more about what hash functions do later
39  * the meantime, think of it as a function that scans
40  * of the input and produces a number.
41  *
42  * For those of you who are more mathematically inclined,
43  * treats each character in the input name as a number.
```

But it looks like we're setting hashValue equal to the number that was returned by the nameHash function. What's going on?

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x555555b6595
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

Name	Value	Type
first	"Ada"	std::string
returned value	1967457	int

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Welcome Edit Design Debug Projects Help

```
19 #include "simpio.h"
20 using namespace std;
21
22 /* Prototype for the nameHash function
23  * in main and the nameHash function
24  */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name? ");
29     string last = getLine("What is your last name? ");
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashValue << endl;
34     return 0;
35 }
36
37 /* This is the actual function that computes the hash value.
38  * to talk more about what hash functions do later
39  * the meantime, think of it as a function that scans
40  * of the input and produces a number.
41  *
42  * For those of you who are more mathematically inclined,
43  * treats each character in the input name as a number.
```

This is pretty cool, actually!

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x5555555b6595
2	std::Function_handler<int (>, QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

Name	Value	Type
first	"Ada"	std::string
last		
hashValue	1967457	int

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



Activities Qt Creator

Jan 4 4:02 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Welcome Edit Design Debug Projects Help

```
19 #include "simpio.h"
20 using namespace std;
21
22 /* Prototype for nameHash
23  * in main and
24  */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first;
29     string last = getLine("What is your last name? ");
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashValue;
34     return 0;
35 }
36
37 /* This is the actual function that computes the hash
38  * to talk more about what hash functions do later
39  * the meantime, think of it as a function that scans
40  * of the input and produces a number.
41  *
42  * For those of you who are more mathematically inclined,
43  * treats each character in the input name as a number.
```

What's happened is that we've just returned from nameHash with a value, but since we're going through the program one step at a time, we haven't actually assigned that value to hashValue yet!

Debugger: GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x555555b6595
2	std::Function_handler<int (>, QtGui::startBackgroundEve...			0x5555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

Name	Value	Type
first	"Ada"	std::string
returned value	1967457	int

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



Activities Qt Creator

Jan 4 4:02 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Welcome Edit Design Debug Projects Help

```
19 #include "simpio.h"
20 using namespace std;
21
22 /* Prototype for nameHash()
23  * in main and
24  */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name? ");
29     string last = getLine("What is your last name? ");
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashValue << endl;
34     return 0;
35 }
36
37 /* This is the actual function that computes the hash value.
38  * to talk more about what hash functions do later
39  * the meantime, think of it as a function that scans
40  * of the input and produces a number.
41  *
42  * For those of you who are more mathematically inclined,
43  * treats each character in the input name as a number.
```

Let's do a "Step Over" so that we can finish executing this line. Click "Step Over," and if you did everything right...

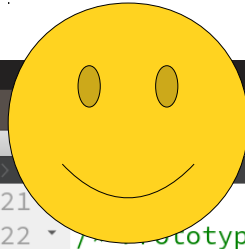
returned value 1967457 int

Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x555555b6595
2	std::Function_handler<int (), QtGui::startBackgroundThread>::operator()			0x5555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

Type to locate (Ctrl+K) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



Activities Qt Creator

Jan 4 4:07 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```
/* Prototype for nameHash function. This lets u
 * in main and then de... in the program.
 */
int nameHash(s

int main() {
    string fir
    string las

    int hashValue = nameHash(first, last);

    cout << "The hash of your name is: " << hashVal
    return 0;
}

/* This is the actual function that computes the ha
 * to talk more about what hash functions do later
 * the meantime, think of it as a function that scr
 * of the input and produces a number.
 *
 * For those of you who are more mathematically inc
 * treats each character in the input name as a num
 * It then uses them as coefficients in a polynomial
 * F_p, where p is a large prime number, and evalu
```

first "Ada" std::string
hashValue 1967457 int
last "Loveland" std::string

... you should see the right value get stored (notice it's in red!) and we've moved to the next line.

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	33	0x555555b65b3
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



Activities Qt Creator

Jan 4 4:07 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21

22 */* Prototype for*

23 ** in main and*

24 **/*

25 `int nameHash(s`

26

27 `int main() {`

28 `string first`

29 `string last = get`

30

31 `int hashValue = nameHash(first, last);`

32

33 `cout << "The hash of your name is: " << hashVal`

34 `return 0;`

35 `}`

36

37 */* This is the actual function that computes the ha*

38 ** to talk more about what hash functions do later*

39 ** the meantime, think of it as a function that scr*

40 ** of the input and produces a number.*

41 ***

42 ** For those of you who are more mathematically inc*

43 ** treats each character in the input name as a num*

44 ** It then uses them as coefficients in a polynomial*

45 ** F_p , where p is a large prime number, and evalu*

At this point, we've seen just about everything we care about. Rather than single-stepping all the way to the end, let's just tell the program to keep on running.

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	33	0x555555b65b3
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 4:07 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashVal
34     return 0;
35 }
36
37 /* This is the actual function that computes the h
38 * to talk more about what hash functions do later
39 * the meantime, think of it as a function that sc
40 * of the input and prod
```

Variable Inspector:

Name	Value	Type
first	"Ada"	std::string
hashValue	1967457	int
last	"Lovelace"	std::string

Debugger Console:

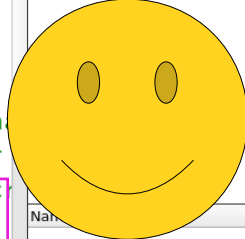
Stopped: "end-stepping-range".

Views: Funct File Line Address Condition Ignore Threads

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

To do this, click on this button. If you hover over it, it says "Continue," and that button means "unpause the program and let it keep running from here."



Activities NameHash Jan 4 4:08 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21

22 */* Prototype for the nameHash function. This lets u*

23 ** in main and then define it later in the program.*

24 **/*

25

26

27

28

29 What is your first name? **Ada**

30 What is your last name? **Lovelace**

31 The hash of your name is: 1967457

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

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458

1459

1460

1461

1462

1463

1464

1465

1466

1467

1468

1469

1470

1471

1472

1473

1474

1475

1476

1477

1478

1479

1480

1481

1482

1483

1484

1485

1486

1487

1488

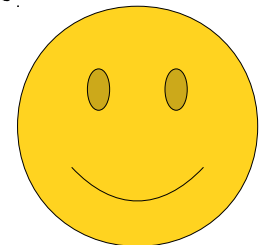
1489

1490

1491

1492

so there you have it! You've now gotten more familiar with the debugger!



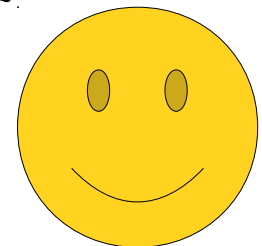
You know how to set a breakpoint to pause the program at a particular point.



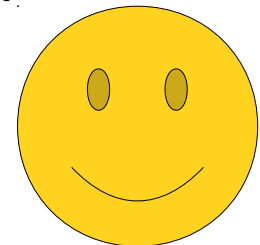
You know how to read the call stack and to see the values of local variables.



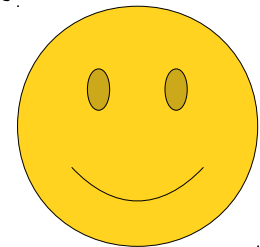
You know how to single-step the program and see what values change.



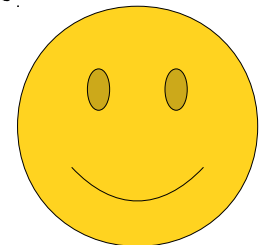
You know how to run a function to completion,
and how to let the program keep on running.



As you write more and more complicated programs this quarter, you'll get a lot more familiar using the debugger and seeing how your programs work.



And, if you continue to build larger and larger pieces of software, you'll find that knowing how to use a debugger is a surprisingly valuable skill!



Hope this helps, and welcome to CS106B!

