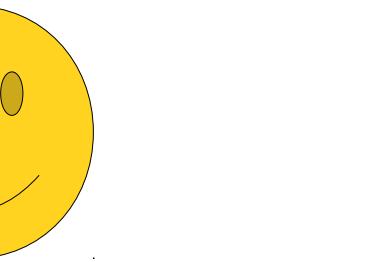


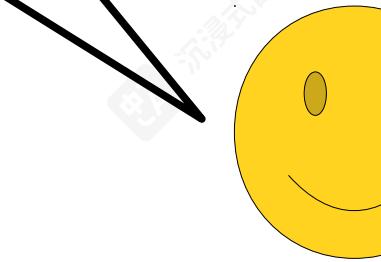
CS106B Debugger Tutorial

CS106B 调试器教程

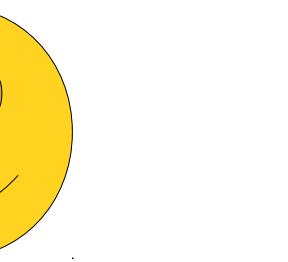
Hi everybody!



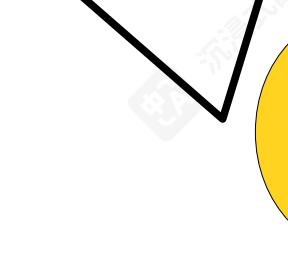
大家好！



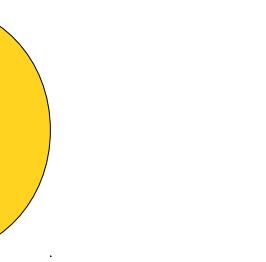
As part of Assignment 0, we'd like you to get a little bit of practice using the debugger in Qt Creator.



作为作业 0 的一部分，我们希望你能稍微练习一下在 Qt Creator 中使用调试器。

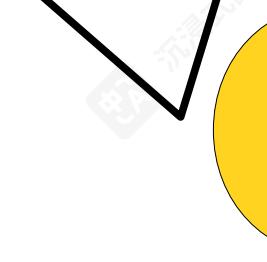


The debugger is a tool you can use to help see what your program is doing as you run it.

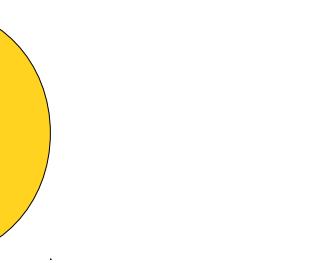


调试器是一个你可以用来帮助查看你的程序在你

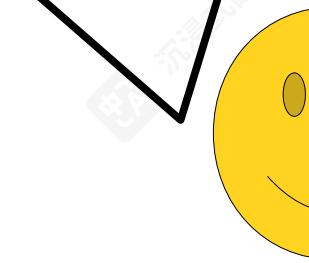
运行它时在做什么。



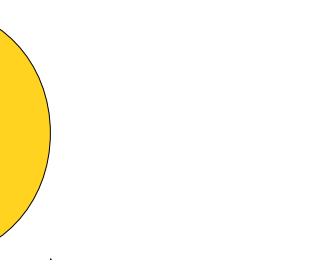
It's really useful for helping find errors in your programs, and the more practice you get with it, the easier it'll be to correct mistakes in the programs you write.



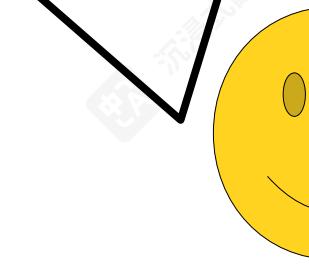
它对于帮助你在程序中查找错误非常有用，而且你越熟悉它，纠正你编写的程序中的错误就越容易。



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.



将此指南视为一个小型教程向导，以帮助你了解如何使用调试器以及如何理解你所看到的内容。



NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

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.

首先, 打开你在本次作业第一部分运行的Name Hash程序。向下滚动到 `nameHash` 函数, 以便 can see the entire function in your window.

```
42 * For those of you  
43 * treats each character as a prime.  
44 * It then multiplies the current hash value by  
45 * F_p, where F_p is the character's index in the  
46 * some small prime number.  
47 * but we treat it like a large prime.  
48 */  
49 int nameHash(const string& first, const string& last)  
50 {  
51     /* This function takes two strings, first and last,  
52      * prime numbers. It returns a hash value where  
53      *  $2^{31} - 1$  is used as the modulus.  
54      */  
55     static const int kLargePrime = 15485863;  
56     static const int kSmallPrime = 137;  
57  
58     int hashVal = 0;  
59  
60     /* Iterate across all the characters in the first name, then the last  
61      * name, updating the hash at each step.  
62      */  
63     for (char ch: first + last) {  
64         /* Convert the input character to lower case. The numeric values of  
65          * lower-case letters are always less than 127.  
66          */  
67         ch = tolower(ch);  
68         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
69     }  
70  
71     return hashVal;
```

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects Projects Projects

Welcome Welcome Welcome

Edit Edit Edit

Design Design Design

Debug Debug Debug

Projects Projects Projects

Help Help Help

Projects Projects Projects

Projects Projects Projects

NameHash [main] NameHash [main] NameHash [main]

NameHash.pro NameHash.pro NameHash.pro

Sources Sources Sources

NameHash.cpp NameHash.cpp 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. These numbers were chosen because their product is less than

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. These numbers were chosen because their product is less than

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

Now, click the mouse!

the last

The numeric values of

for (char ch: first + last) {
/* Convert the input character to
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargeP

the last

The numeric values of

for (char ch: first + last) {
/* Convert the input character to
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargeP

return hashVal;

return hashVal;

Line: 11, Col: 48 Line: 11, Col: 48

Type to locate (Ctrl...) Type to locate (Ctrl...)

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

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 * the character's position in the name. You can see this for CS106B,

When you do, you should see a red circle with a little hourglass pop up.

This is called a **breakpoint**. If we run the program in debug mode, whenever the program gets to this line, it will pause and open up the debugger so we can see what's going on.

The numeric values of the last character in the name are always less than 127.

当你这样做时，你应该看到一个带有一个小沙漏弹出的红圈。

这被称为一个 **断点**。如果我们运行程序在调试模式下，每当程序到达这一行时，它将暂停并打开调试器这样我们就能看到发生了什么。

the last

the last

61
62 for (char ch: first + last) {
63 /* Convert the input character to
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargeP
68 }
69
70 return hashVal;

61
62 for (char ch: first + last) {
63 /* Convert the input character to
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargeP
68 }
69
70 return hashVal;



NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects Projects Projects

Welcome Welcome Welcome

Edit Edit Edit

Design Design Design

Debug Debug Debug

Projects Projects Projects

Help Help Help

Projects Projects Projects

Projects Projects Projects

NameHash [main] NameHash [main] NameHash [main]

NameHash.pro NameHash.pro NameHash.pro

Sources Sources Sources

NameHash.cpp NameHash.cpp 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. These numbers were chosen because their product is less than
```

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...

the last

```
62 for (char ch: first + last) {
63     /* Convert the input character to
64     * lower-case letters are always less than 127.
65     */
66     ch = tolower(ch);
67     hashVal = (kSmallPrime * hashVal + ch) % kLargeP
68 }
69
70 return hashVal;
```

The numeric values of

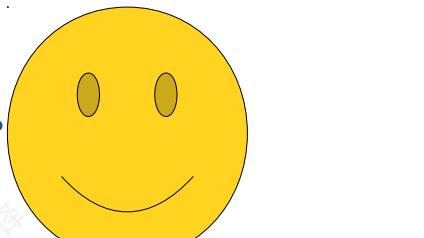


the last

the last

```
62 for (char ch: first + last) {
63     /* Convert the input character to
64     * lower-case letters are always less than 127.
65     */
66     ch = tolower(ch);
67     hashVal = (kSmallPrime * hashVal + ch) % kLargeP
68 }
69
70 return hashVal;
```

The numeric values of



the last

现在，我们将以调试模式运行此程序
为此，请点击屏幕右下角的“以调试模式运行”
屏幕右下角的按钮。
它是常规绿色“运行”按钮正下方的那个。
当你这样做时...

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 the
43 * treats
44 * It the
45 * F_p, w
46 * some s
47 * but we
48
49 File Edit Options Help
50
51 What is your first name? |
52
53
54
55
56
57
58
59
60
ame, then
61
62
ame, then
63
he numeric
64 * lower-case letters are always less than 127.
65 */

Debugger GDB for "NameHash" Application started.

Threads: #1

Level Function File Number Function File Line Address Con Ignore Threads

1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash [main] NameHash.pro Sources NameHash.cpp

42 * For the
43 * treats
44 * It the
45 * F_p, w
46 * some s
47 * but we
48
49 File Edit Options Help
50
51 What is your first name? |
52
53
54
55
56
57
58
59
ame, then
60
61
ame, then
62
63
he numeric
64 * lower-case letters are always less than 127.
65 */

Debugger GDB for "NameHash" Application started.

Threads: #1

Level Function File Number Function File Line Address Con Ignore Threads

1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

... 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.

...你应该看到类似这样的内容！注意在Qt Creator中弹出了许多额外的面板。我们稍后会讨论每个这些窗口的含义一秒钟后。

NameHash.cpp @ NameHash [main] - Qt Creator

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.)

此时，输入第一个名字 **Ada** 并按下输入，如这里所示。我们特别希望您 enter **Ada** here, *不是你的真实名字*。
(除非你的名字是Ada。)

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

ame, then
he numeric

* lower-case letters are always less than 127.
*/

Debugger GDB for "NameHash" Application started.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Debug Projects Help

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

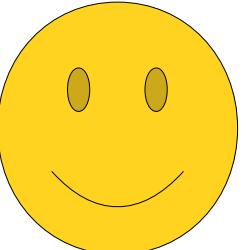
42 * For th
43 * treats
44 * It the
45 * F_p, w
46 * some s
47 * but we
48
49 File Edit Options Help
50
51 What is your first name? Ada
52 What is your last name? Lovelace|
53
54
55
56
57
58
59
60
61
62
63
64 * lower-case letters are always less than 127.
65 */

Debugger GDB for "NameHash" Application started.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

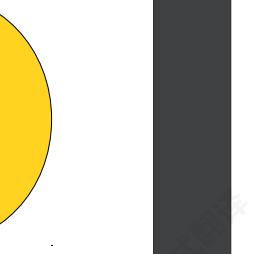
1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results



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

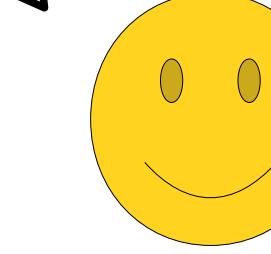
1, type in **Lovelace** as a last name, but
don't hit enter yet!



The screenshot shows the Qt Creator IDE interface. The top menu bar includes File, Edit, View, Build, Debug, Analyze, Tools, Window, and Help. The Projects panel on the left shows a project named "NameHash [main]" with files NameHash.pro and NameHash.cpp selected. The main area displays a terminal window with the following text:
What is your first name? Ada
What is your last name? Lovelace

A large callout bubble with a black border and a purple outline contains the text: "现在, 输入 Lovelace 作为姓氏, 但不要按回车键!" (Now, enter Lovelace as the last name, but don't press Enter!). A yellow smiley face icon is positioned to the right of the terminal window.

现在，输入 **Lovelace** 作为
不要按回车键！



NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

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

42 * For the
43 * treats
44 * It the
45 * F_p, w
46 * some s
47 * but we
48
49 File Edit Options Help
50
51 What is your first name? Ada
52 What is your last name? Lovelace
53
54
55
56
57
58
59
ame, then
60
61
62
63
ame, then
64 * lower-case letters are always less than 127.
65 */

File Edit Options Help
50
51 What is your first name? Ada
52 What is your last name? Lovelace
53
54
55
56
57
ame, then
58
59
60
61
62
63
ame, then
64 * lower-case letters are always less than 127.
65 */

Debugger GDB for "NameHash" Application started.

Debugger GDB for "NameHash" Application started.

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

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

一旦按下回车键，就会发生很多事情。即将在 Qt Creator 中弹出。别慌！

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

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

既然如此，按回车键，
然后见证奇迹发生！

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

ame, then
he numeric

* lower-case letters are always less than 127.

*/

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Debug Projects ? Help

Welcome Edit Design Debug Projects ? Help

Debugger GDB for "NameHash" Threads: #1 Application started.

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Debugger GDB for "NameHash" Threads: #1 Application started.

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

沉浸式翻译

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Debug Projects NameHash [main] NameHash.pro Sources NameHash.cpp

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 1548563; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137
56
57
58
59 /* Iterates over each character in the name
60 * name.
61 */
62 for (char ch : name)
63 {
64 /* Converts character to lowercase
65 * before adding to hash value.
66 */
67 ch = tolower(ch);
68 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69 }
70
71 return hashVal;
72 }

Line: 66, Col: 9

Name Value Type
_for_begin @0x7ffffb2ffcb78 std::string::iterator
_for_end @0x7ffffb2ffcb80 std::string::iterator
_for_range ch "AdaLovelace" std::string &
first 'A' 65 char
hashVal 0 std::string int
kLargePrime 1548563 int
kSmallPrime 137 int
last "Lovelace" std::string

0x41

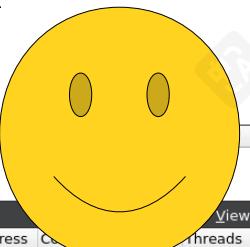
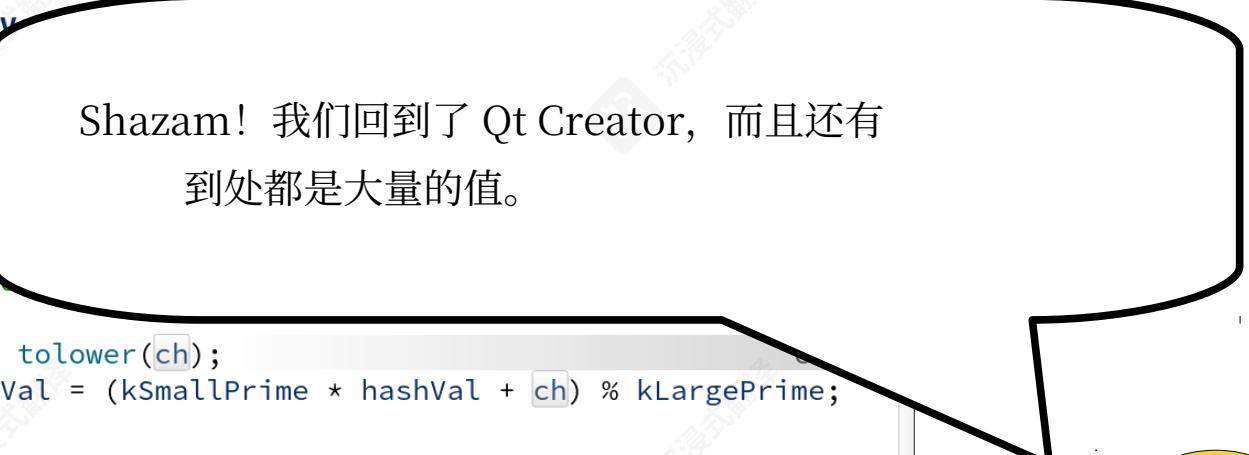
Shazam! We're back in Qt Creator, and there's tons of values showing up everywhere.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15. Level Function File Line Address Number Function File Line Address Callers
1 nameHash NameHash.cpp 66 0x5555555ab2d3 1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3
2 studentMain NameHash.cpp 31 0x5555555ab0fb
3 std::Function_ha... 0x5555556037fc
4 GThreadStd::run() 0x5555555e6616
5 ?? 0x7ffff64dc2b3
6 start_thread pthread_create.c 442 0x7ffff6094b43
7 clone3 clone3.S 81 0x7ffff6126a00
Views

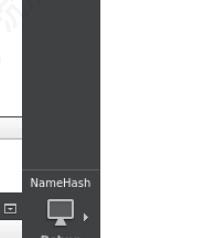


沉浸式学习

Shazam! 我们回到了 Qt Creator, 而且还有到处都是大量的值。



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



沉浸式学习

这里发生了很多事情。让我们看看发生了什么。



NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137

56
57
58
59
60
61
62
63
64
65
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69
70 return hashVal;
71
72 }

Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Co Threads (all)

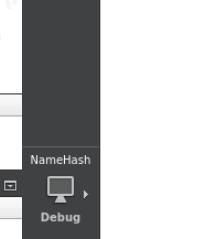
Level	Function	File	Line	Address	Number	Function	File	Line	Address	Co	Threads
1	nameHash	NameHash.cpp	66	0x555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3		(all)
2	studentMain	NameHash.cpp	31	0x5555555ab0fb							
3	std::_Function_h...			0x555555603fc							
4	GThreadStd::run()			0x5555555e6616							
5	??			0x7ffffe4dc2b3							
6	start_thread	pthread_create.c	442	0x7ffff6094b43							
7	clone3	clone3.S	81	0x7ffff6126a00							

Type to locate (Ctrl) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

生了很多事情。让我们
发



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



首先, 请注意我们的红色断点现在有一个
黄色的箭头在里面。

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;    kLargePrime: 1...
static const int kSmallPrime = 137;          kSmallPrime: 137

int hashVal(const std::string& name) {
    /* Iterates over each character in the string */
    for (char ch : name) {
        /* Converts character to lowercase */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

调试器显示: 断点已设置在第 15 行。线程: #15 NameHash

Level	Function	File	Line	Address	Number	Function	File	Line
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66
2	studentMain	NameHash.cpp	31	0x5555555ab0fb				
3	std::_Function_han			0x55555556037fc				
4	GThreadStd::run()			0x55555555e6616				
5	??			0x7ffff64dc2b3				
6	start_thread	pthread_create.c	442	0x7ffff6094b43				
7	clone3	clone3.S	81	0x7ffff6126a00				

請注意我們的紅色斷點現
黃色的箭頭在



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.

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137
56
57
58
59
60
61
62
63
64
65
66 ch = tolower(ch); hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; ▶
67 }
68
69
70
71
72 }

/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.

int hashV
/* Iterat
* name
*/
for (ch
/* C

Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

NameValueType
_for_begin@0x7fffb2ffcb78std::string::iterator
_for_end@0x7fffb2ffcb80std::string::iterator
_for_range@0x41std::string &
ch'A'char
first"AdaLovelace"std::string
hashVal0int
kLargePrime15485863int
kSmallPrime137int
last"AdaLovelace"std::string

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Projects Help

NameHash.cpp @ NameHash [main] - Qt Creator

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

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137

56
57
58
59
60
61
62
63
64
65 int hashV
66 /* Iterat
67 * name
68 */
69 for (char ch : name)
70 /* C
71 * To
72 */
73 ch = tolower(ch);
74 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

75
76
77
78
79
80
81 return hashVal;

Line: 66, Col: 9

Name Value Type
for_begin @0x7ffffb2ffcb78 std::string::iterator
for_end @0x7ffffb2ffcb80 std::string::iterator
for_range "AdaLovelace" std::string
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

0x41

这个黄色箭头指示程序中的位置
我们目前所在的位置。程序在
这一行停止运行，因为我们触发了你设置的断点
之前设置的

之前设置的

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Threads (all)

Level	Function	File	Line	Address	Number	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555ab2d3	1	nameHash(std::string, std::string)	..eHash.cpp	66	
2	studentMain	NameHash.cpp	31	0x555555ab0fb					
3	std::_Function_ha...			0x5555556037fc					
4	GThreadStd::run()			0x5555555e6616					
5	??			0x7ffff64dc2b3					
6	start_thread	pthread_create.c	442	0x7ffff6094b43					
7	clone3	clone3.S	81	0x7ffff6126a00					

Type to locate (Ctrl) 1. Issues 2. Search Results 3. Application Output 4. Compile Output 5. QML Debugger Console 6. General Messages 7. Version Control 8. Test Results



NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 66, Col: 9

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 66, Col: 9

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 66, Col: 9

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.
 */

static const int kLargePrime = 15485863; kLargePrime: 15485863
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal;
/* Iterators */
for (char ch = 'A'; ch <= 'Z'; ch++) {
 /* Convert character to lowercase */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;

0x41 char std::string

Name _for_begin @0x7fff2ffcb78
Value @0x7fff2ffcb80
Type std::string::iterator

_for_end @0x7fff2ffcb80
Value "AdaLovelace"
Type std::string::iterator

_for_range std::string &

ch 'A' 65
first "Ada" 65

hashVal 0
kLargePrime 15485863
kSmallPrime 137

last "Lovelace"

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.
 */

static const int kLargePrime = 15485863; kLargePrime: 15485863
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal;
/* Iterators */
for (char ch = 'A'; ch <= 'Z'; ch++) {
 /* Convert character to lowercase */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;

0x41 char std::string

Name _for_begin @0x7fff2ffcb78
Value @0x7fff2ffcb80
Type std::string::iterator

_for_end @0x7fff2ffcb80
Value "AdaLovelace"
Type std::string::iterator

_for_range std::string &

ch 'A' 65
first "Ada" 65

hashVal 0
kLargePrime 15485863
kSmallPrime 137

last "Lovelace"

每当您弹出调试器时，最好
弄清楚您在程序中的确切位置
您正在运行，这样您就会养成
检查这个黄色箭头的习惯。

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.

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

```
File Edit View Build Debug Analyze Tools Window Help
```

```
File Edit View Build Debug Analyze Tools Window Help
```

Projects NameHash.cpp nameHash(string, string) -> int Line: 66, Col: 9

Projects NameHash.cpp nameHash(string, string) -> int Line: 66, Col: 9

```
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
```

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. Then we can
     * lower
     */
    ch = t
    hashVa
}
```

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

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. Then we can
     * lower
     */
    ch = t
    hashVa
}
```

接下来，让我们看看这个面板。
这被称为 **调用栈**。

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0f9								
3	std::Function_h...			0x5555556037fc								
4	GThreadStd::run()			0x5555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7fff6094b43								
7	clone3	clone3.S	81	0x7fff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. Then we can
     * lower
     */
    ch = t
    hashVa
}
return has
```

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!

Name Value Type

Name	Value	Type
_for_begin	@0x7ffb2ffcb78	std::string::iterator
_for_end	@0x7ffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. Then we can
     * lower
     */
    ch = t
    hashVa
}
return has
```

此时，我们知道我们处于 `nameHash` function, because our helpful friend the Yellow Arrow tells us exactly what line we're on!

Name Value Type

Name	Value	Type
_for_begin	@0x7ffb2ffcb78	std::string::iterator
_for_end	@0x7ffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. Then we can
     * lower
     */
    ch = t
    hashVa
}
return has
```

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. Then we can
     * lower
     */
    ch = t
    hashVa
}
return has
```

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::Function_h...			0x5555556037fc								
4	GThreadStd::run()			0x555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7fff6094b43								
7	clone3	clone3.S	81	0x7fff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

/* This hashing scheme needs two prime numbers, a large prime.
 * prime. These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.

*/

static const int **kLargePrime** = 15485863; kLargePrime: 1...

static const int **kSmallPrime** = 137; kSmallPrime: 137

int **hashVal** = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.

*/

for (char **ch**: first + last) { first: "Ada" last: "Love..."
 /* Convert the input character to lower case. Then
 * lower
 */
 ch = **t**
 hashVa

}

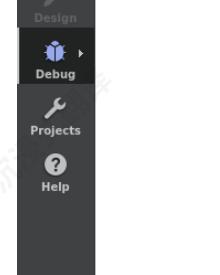
return has

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`?

Name	Value	Type
_for_begin	@0x7ffb2ffcb78	std::string::iterator
_for_end	@0x7ffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con/Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)	
2	studentMain	NameHash.cpp	31	0x5555555ab0fb							
3	std::_Function_han			0x55555556037fc							
4	GThreadStd::run()			0x55555555e6616							
5	??			0x7ffff64dc2b3							
6	start_thread	pthread_create.c	442	0x7ffff6094b43							
7	clone3	clone3.S	81	0x7ffff6126a00							



/* This hashing scheme needs two prime numbers, a large prime.
 * prime. These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.
 */

```
static const int kLargePrime = 15485863;    kLargePrime: 1...
```

```
static const int kSmallPrime = 137;          kSmallPrime: 137
```

```
int hashVal = 0;                           hashVal: 0
```

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */

```
for (char ch: first + last) {      first: "Ada"  last: "Love..."
```

/* Convert the input character to lower case. Then
 * lower
 */

```
ch = t  
hashVa
```

}

return has

Line: 66, Col: 9

Name	Value
__for_begin	@0x7ffb2ffcb78
__for_end	@0x7ffb2ffcb80
__for_range	"AdaLovelace"
ch	'A'
first	"Ada"
hashVal	0
kLargePrime	15485863
kSmallPrime	137
last	"Lovelace"



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`?

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`?

File Edit View Build Debug Analyze Tools Window Help

NameHash.cpp @ NameHash [main] - Qt Creator

Projects NameHash [main] Sources NameHash.cpp

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137
56
57 int hashVal = 0; hashVal: 0
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) { first: "Ada" last: "Love...
63 /* Convert the input character to lower case. The reason
64 * lower
65 */
66 ch = t
67 hashVa
68 }
69
70 return has
71
72 }

Line: 66, Col: 9

Name Value Type
__for_begin @0x7fff2ffcb78 std::string::iterator
__for_end @0x7fff2ffcb80 std::string::iterator
__for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

0x41

然，黄色箭头无法告诉我们确切
我们是如何到达程序的这个部分的。什么
程序的部分实际上调用了nameHash?

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 66 0x5555555ab2d3 1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

2 studentMain NameHash.cpp 31 0x5555555ab0fb
3 std::Function::ha... 0x55555556037fc
4 GThreadStd::run() 0x5555555e616
5 ?? 0x7ffff64dc2b3
6 start_thread pthread_create.c 442 0x7ffff6094b43
7 clone3 clone3.S 81 0x7ffff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results



而，黄色箭头无法告诉我们确定我们是如何到达程序的这个部分。程序的部分实际上调用了name

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

```
File Edit View Build Debug Analyze Tools Window Help
```

```
File Edit View Build Debug Analyze Tools Window Help
```

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 66, Col: 9

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 66, Col: 9

```
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
```

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. Then we can
     * lower
     */
    ch = t
    hashVa
}

return has
```

/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
 /* Convert the input character to lower case. Then we can
 * lower
 */
 ch = t
 hashVa
}

return has

The call stack can tell us exactly that!

调用栈可以告诉我们确切的情况!

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::Function_h...			0x5555556037fc								
4	GThreadStd::run()			0x5555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7fff6094b43								
7	clone3	clone3.S	81	0x7fff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

```
File Edit View Build Debug Analyze Tools Window Help
```

```
File Edit View Build Debug Analyze Tools Window Help
```

Projects NameHash.cpp Line: 66, Col: 9

Projects NameHash.cpp Line: 66, Col: 9

```
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
```

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. Then we can
     * lower
     */
    ch = t
    hashVa
}
```

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`.

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555ab2d3
2	studentMain	NameHash.cpp	31	0x555555ab0fb
3	std::Function_h...			0x5555556037fc
4	GThreadStd::run()			0x555555e616
5	??			0x7fff64dc2b3
6	start_thread	pthread_create.c	442	0x7fff6094b43
7	clone3	clone3.S	81	0x7fff6126a00

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555ab2d3
2	studentMain	NameHash.cpp	31	0x555555ab0fb
3	std::Function_h...			0x5555556037fc
4	GThreadStd::run()			0x555555e616
5	??			0x7fff64dc2b3
6	start_thread	pthread_create.c	442	0x7fff6094b43
7	clone3	clone3.S	81	0x7fff6126a00

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x7fff2ffcb78
2	studentMain	NameHash.cpp	31	0x7fff2ffcb80
3	std::Function_h...			"AdaLovelace"
4	GThreadStd::run()			'A' 65
5	??			"Ada"
6	hashVal			0
7	kLargePrime			15485863
8	kSmallPrime			137
9	last			"Lovelace"

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x7fff2ffcb78
2	studentMain	NameHash.cpp	31	0x7fff2ffcb80
3	std::Function_h...			"AdaLovelace"
4	GThreadStd::run()			'A' 65
5	??			"Ada"
6	hashVal			0
7	kLargePrime			15485863
8	kSmallPrime			137
9	last			"Lovelace"

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4	GThreadStd::run()			0x41
5	??			0x41
6	start_thread	pthread_create.c	442	0x41
7	clone3	clone3.S	81	0x41

Call Stack:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x41
2	studentMain	NameHash.cpp	31	0x41
3	std::Function_h...			0x41
4				

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Debug Projects Help

NameHash.cpp @ NameHash [main] - Qt Creator

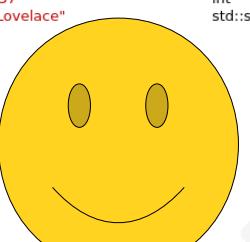
Projects NameHash [main] Sources NameHash.cpp

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137
56
57 int hashVal = 0; hashVal: 0
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) { first: "Ada" last: "Love...
63 /* Convert the input character to lower case. The n...
64 * lower
65 */
66 ch = t
67 hashVa
68 }
69
70 return has

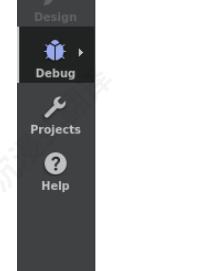
Line: 66, Col: 9

Name Value Type
_for_begin @0x7ffb2ffcb78 std::string::iterator
_for_end @0x7ffb2ffcb80 std::string::iterator
_for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string int
hashVal 0 std::string int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string int

0x41



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



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

File Edit View Build Debug Analyze Tools Window Help

NameHash.cpp @ NameHash [main] - Qt Creator

Projects NameHash [main] Sources NameHash.cpp

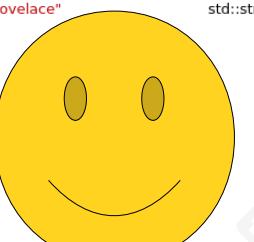
50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137
56
57 int hashVal = 0; hashVal: 0
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) { first: "Ada" last: "Love...
63 /* Convert the input character to lower case. The range
64 * lower bound is 'A' and upper bound is 'Z'.
65 */
66 ch = tolower(ch); ch: 'A'
67 hashVal = (hashVal * kSmallPrime + ch) % kLargePrime;
68 }
69
70 return hashVal;

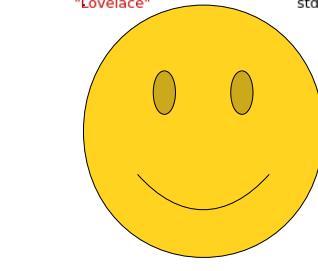
Line: 66, Col: 9

Name Value Type
__for_begin @0x7fff82ffcb78 std::string::iterator
__for_end @0x7fff82ffcb80 std::string::iterator
__for_range "Ada" 0x41 char
ch 'A' 65 std::string::iterator
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

0x41

去并双击调用 studentMain
在 2 级。当你做...





调用 student
在 2 级。当

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

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

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

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

```
using namespace std;  
  
/* Prototype for the nameHash function. This lets us use the function  
 * 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);    hashValue: 0  fi...  
  
    cout << "The hash of your name is: " << hashValue << endl;  
    return 0;  
}  
  
/* This is the actual function that does the hashing.  
 * To talk more about what hash functions do,  
 * the meantime, think of it as a function  
 * of the input and produces a number.  
 */  
  
/* For those of you who are more mathematically inclined, this function  
 * treats each character in the input name as a number between 0 and 255.  
 */
```

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

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

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

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

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

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

You'll end up over here!

您最终会在这里 !

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

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

```
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets us use the function
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);    hashValue: 0 fi...
32
33     cout << "The hash of your name is: "
34     return 0;
35 }
36
37 /* This is the actual function that does all the work.
38 * To talk more about what hash functions do in the meantime,
39 * think of it as a function that takes some input
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 as a number
44 * and adds them together to get the hash value.
```

Line: 31, Col: 5

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

unix (LF)

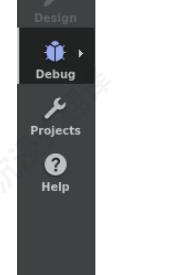
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets us use the function
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28 string first = getLine("What is your first name? ");
29 string last = getLine("What is your last name? ");
30
31 int hashValue = nameHash(first, last); hashValue: 0 fi...
32
33 cout << "The hash of your name is: "
34 return 0;
35 }
36
37 /* This is the actual function that does all the work.
38 * To talk more about what hash functions do in the meantime,
39 * think of it as a function that takes some input
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 as a number
44 * and adds them together to get the hash value.

Debugger GDB for "NameHash" Threads: #15 NameHash

Level	Function	File	Line	Address	Number	Function
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	1	nameHash(std::s...
2	studentMain	NameHash.cpp	31	0x5555555ab0fb		
3	std::Function::ha...			0x5555556037fc		
4	GThreadStd::run()			0x5555555e6616		
5	??			0x7ffff64dc2b3		
6	start_thread	pthread_create.c	442	0x7ffff6094b43		
7	clone3	clone3.S	81	0x7ffff6126a00		

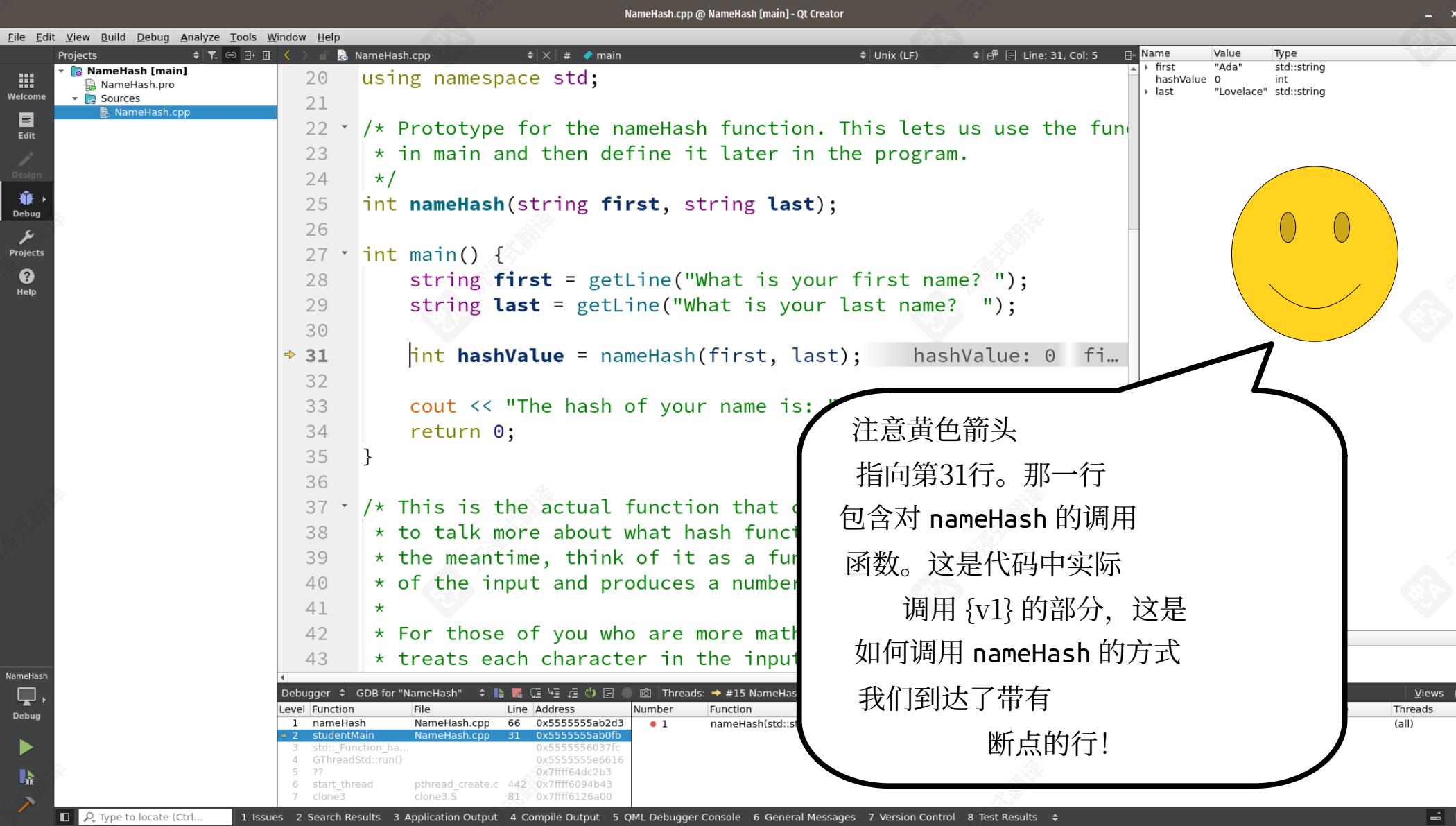
Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

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!

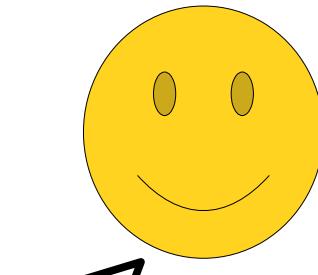


Notice that the yellow arrow points to Line 31. That line includes a call to the `nameHash` function. This is the part of the code that actually called `nameHash`, which is how we got to the line with the breakpoint!

注意黄色箭头
指向第31行。那一行
包含对 `nameHash` 的调用
函数。这是代码中实际
调用 {vl} 的部分，这是
如何调用 `nameHash` 的方式
我们到达了带有
断点的行！



The screenshot shows the Qt Creator IDE interface. The main window displays a C++ file named `NameHash.cpp`. The code defines a `nameHash` function and its prototype. It prompts the user for their first and last names, then prints the hash value. A yellow smiley face icon is positioned in the top right corner. A large black callout bubble points from the bottom right towards the line number 31 in the code editor. The status bar at the bottom shows the debugger information: "Debugger GDB for 'NameHash'". The bottom left contains navigation icons for issues, search results, application output, compile output, QML debugger console, general messages, version control, and test results.



注意黄色箭头
指向第31行。那一行
包含对 `nameHash` 的调用。
函数。这是代码中实际
调用 `{v1}` 的部分，
如何调用 `nameHash` 的
我们到达了带有
断点的行！

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

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

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

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

```
using namespace std;  
  
/* Prototype for the nameHash function. This lets us use the function  
 * 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);    hashValue: 0  fi...  
  
    cout << "The hash of your name is: " << hashValue << endl;  
    return 0;  
}  
  
/* This is the  
 * to talk more  
 * the meantime  
 * of the input  
 *  
 * For those of you who are more mathematically inclined, this function  
 * treats each character in the input name as a number between 0 and 255.  
 */
```

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!

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

Name Value Type

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

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

Name Value Type

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

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	● 1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb	2	studentMain	NameHash.cpp	31	0x5555555ab0fb			
3	std::function<_hasher>::operator()		0x55555556037fc		3	std::function<_hasher>::operator()		0x55555556037fc				
4	GThreadStd::run()		0x5555555e616		4	GThreadStd::run()		0x5555555e616				
5	??		0x7ffff64dc2b3		5	??		0x7ffff64dc2b3				
6	start_thread	pthread_create.c	442	0x7ffff6094b43	6	start_thread	pthread_create.c	442	0x7ffff6094b43			
7	clone3	clone3.S	81	0x7ffff6126a00	7	clone3	clone3.S	81	0x7ffff6126a00			

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

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

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

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

```
using namespace std;  
  
/* Prototype for the nameHash function. This lets us use the function  
 * 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);    hashValue: 0  fi...  
  
    cout << "The hash of your name is: " << hashValue << endl;  
    return 0;  
}  
  
/* This is the  
 * to talk more  
 * the meantime  
 * of the input  
 *  
 * For those of you who are more mathematically inclined, this function  
 * treats each character in the input name as a number between 0 and 255.
```

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

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

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

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

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

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

Depending on your OS, you might see some additional functions beneath studentMain. What are those?

根据您的操作系统，您可能会看到一些在 studentMain 下方的一些附加功能。那些是什么？

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb	2	studentMain	NameHash.cpp	31	0x5555555ab0fb			
3	std::Function_h...		0x55555556037fc		3	std::Function_h...		0x55555556037fc				
4	GThreadStd::run()		0x5555555e616		4	GThreadStd::run()		0x5555555e616				
5	???		0x7fff64dc2b3		5	???		0x7fff64dc2b3				
6	start_thread	pthread_create.c	442	0x7ffff6094b43	6	start_thread	pthread_create.c	442	0x7ffff6094b43			
7	clone3	clone3.S	81	0x7ffff6126a00	7	clone3	clone3.S	81	0x7ffff6126a00			

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

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

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

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

```
using namespace std;  
  
/* Prototype for the nameHash function. This lets us use the function  
 * 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);    hashValue: 0  fi...  
  
    cout << "The hash of your name is: " << hashValue << endl;  
    return 0;  
}  
  
/* This is the  
 * to talk more  
 * the meantime  
 * of the input  
 *  
 * For those of you who are more mathematically inclined, this function  
 * treats each character in the input name as a number between 0 and 255.  
 */
```

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

Name Value Type

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

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

Name Value Type

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

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

这些灰色函数表示辅助函数
我们的库自动调用以帮助
让你的程序设置好。

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

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

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

```
using namespace std;
/* Prototype for the nameHash function. This lets us use the function 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);    hashValue: 0  file:///home/.../NameHash.cpp#L31

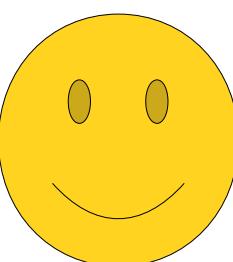
    cout << "The hash of your name is: " << hashValue << endl;
    return 0;
}

/* This is the
 * to talk more
 * the meantime
 * of the input
 *
 * For those of you who are more mathematically inclined, this function
 * treats each character in the input name as a number between 0 and 255.
 */


```

Line: 31, Col: 5

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

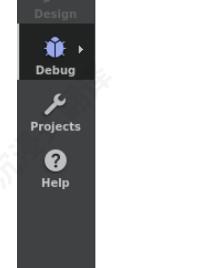


You don't need to worry about these. They'll show up in all the programs you run and you can safely ignore them.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::Function::operator()			0x5555556037fc								
4	GThreadStd::run()			0x55555556e616								
5	??			0x7ffff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

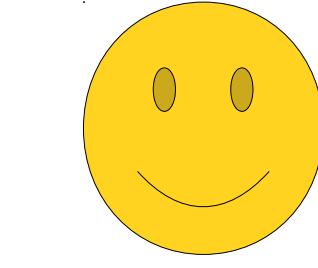


don't need to worry about these. They'll
up in all the programs you run and you can
safely ignore them.

The screenshot shows the Qt Creator IDE interface. The main window displays a C++ file named `NameHash.cpp`. The code defines a `nameHash` function that concatenates two strings and returns their hash value. A callout bubble with a yellow smiley face icon contains the following Chinese text:

您无需担心这些。它们将
出现在您运行的所有程序中，并且您可以
安全地忽略它们。

The code editor also shows a variable table and a debugger window at the bottom.



心这些。它们将
行的所有程序中，并
安全地忽略它

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

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

```
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets us use the function
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);    hashValue: 0 fi...
32
33     cout << "The hash of your name is: " << hashValue << endl;
34     return 0;
35 }
```

Name Value Type

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

Welcome Edit Design Debug Projects Help

Name Hash [main] NameHash.pro Sources NameHash.cpp

```
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets us use the function
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);    hashValue: 0 fi...
32
33     cout << "The hash of your name is: " << hashValue << endl;
34     return 0;
35 }
```

Name Value Type

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

Welcome Edit Design Debug Projects Help

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...

在此期间，让我们回到我们的 **nameHash** 函数。要这样做，请双击调用堆栈顶部的 **nameHash** 条目。当你这样做...

File Edit View Build Debug Analyze Tools Window Help

Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	std::function<int(const std::string&, const std::string&)>::operator()	functional.h	11	0x5555555ab0fb								
3	std::function<int(const std::string&, const std::string&)>::operator()	functional.h	37fc	0x55555556037fc								
4	GThreadStd::GThread::start	gthread.h	616	0x5555555e616								
5	??			0x7ffff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp Line: 66, Col: 9

Projects NameHash.cpp Line: 66, Col: 9

```
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
```

/* This hashing scheme needs two prime numbers, a large prime.
 * prime. These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.
 */

static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */

for (char ch: first + last) { first: "Ada" last: "Love..."
 /* Convert the input character to lower case. Then
 * lower.
 */
 ch = t
 hashVal

You'll be teleported back here!

return has

Name Value Type

_for_begin @0x7ffffb2ffcb78 std::string::iter...
_for_end @0x7ffffb2ffcb80 std::string::iter...
_for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

Name Value Type

_for_begin @0x7ffffb2ffcb78 std::string::iter...
_for_end @0x7ffffb2ffcb80 std::string::iter...
_for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

Name Value Type

_for_begin @0x7ffffb2ffcb78 std::string::iter...
_for_end @0x7ffffb2ffcb80 std::string::iter...
_for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 66 0x5555555ab0d3 1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

2 studentMain NameHash.cpp 31 0x5555555ab0fb 2 studentMain NameHash.cpp 31 0x55555556037fc 3 std::Function_h... 0x55555556037fc 4 GThreadStd::run() 0x5555555e616 5 ?? 0xfffff64dc2b3 6 start_thread pthread_create.c 442 0x7ffff6094b43 7 clone3 clone3.S 81 0x7ffff6126a00 5 ?? 0xfffff64dc2b3 6 start_thread pthread_create.c 442 0x7ffff6094b43 7 clone3 clone3.S 81 0x7ffff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp Line: 66, Col: 9

Projects NameHash.cpp Line: 66, Col: 9

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0

/* Iterate
 * name, u
 */
for (char
    /* Con
     * lowe
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;
```

/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0

/* Iterate
 * name, u
 */
for (char
 /* Con
 * lowe
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

Name Value Type
_for_begin @0x7fffb2ffcb78 std::string::iter...
_for_end @0x7fffb2ffcb80 std::string::iter...
_for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

Name Value Type
_for_begin @0x7fffb2ffcb78 std::string::iter...
_for_end @0x7fffb2ffcb80 std::string::iter...
_for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

0x41

0x41

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

让我们快速回顾一下我们到目前为止所看到的内容。

ch: 65

ch: 65

return hashVal;

return hashVal;

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::Function_h...			0x55555556037fc								
4	GThreadStd::run()			0x5555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7fff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

The screenshot shows the Qt Creator IDE interface with the following details:

- File Menu:** File, Edit, View, Build, Debug, Analyze, Tools, Window, Help.
- Projects:** NameHash [main] (selected), NameHash.pro, Sources, NameHash.cpp.
- Code Editor:** Line 66, Col: 9, code snippet:

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;    kLargePrime: 1...
static const int kSmallPrime = 137;          kSmallPrime: 137

int hashVal = 0

/* Iterate
 * name, u
 */
for (char
/* Con
 * lowe
 */
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;
```
- Breakpoint:** A red arrow points to the line number 66 in the margin, indicating where a breakpoint is set.
- Callout Bubble:** A large black callout bubble contains handwritten text: "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."
- Variables View:** Shows local variables with their values and types. The variable `ch` has a value of 65 and a type of `char`. Other variables include `for_begin`, `for_end`, `for_range`, `first`, `hashVal`, `kLargePrime`, `kSmallPrime`, and `last`.
- Registers View:** Shows registers with their values and types. The register `ch` has a value of 65 and a type of `char`.
- Stack View:** Shows the call stack with the current frame at line 66.
- Output View:** Shows the output of the debugger, including threads and stopped points.
- Help:** A yellow smiley face icon is located in the bottom right corner.

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.



要设置一个断点以便我们可以暂停
程序并查看周围，点击你想要暂停的行号
之前的边缘。

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137

56
57 int hashVal = 0
58
59 /* Iterate
60 * name, u
61 * */
62 for (char
63 /* Con
64 * lowe
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69
70 return hashVal;
71
72 }

Line: 66, Col: 9

Name Value Type
for_begin @0x7fffb2ffcb78 std::string::iterator
for_end @0x7fffb2ffcb80 std::string::iterator
for_range "AdaLovelace" std::string &
ch 'A' char 0x41
first "Ada" std::string int
hashVal 0 std::string int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function

1 nameHash NameHash.cpp 66 0x555555ab2d3 1 nameHash(std::string, std::string)

2 studentMain NameHash.cpp 31 0x555555ab0fb 2 studentMain()

3 std::_Function_han... 0x555555603fc 3 std::_Function_han...()

4 GThreadStd::run() 0x5555555e6616 4 GThreadStd::run()

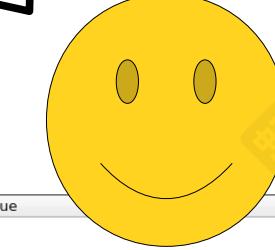
5 ?? 0x7ffffe4dc2b3 5 ??

6 start_thread pthread_create.c 442 0x7ffff6094b43 6 start_thread()

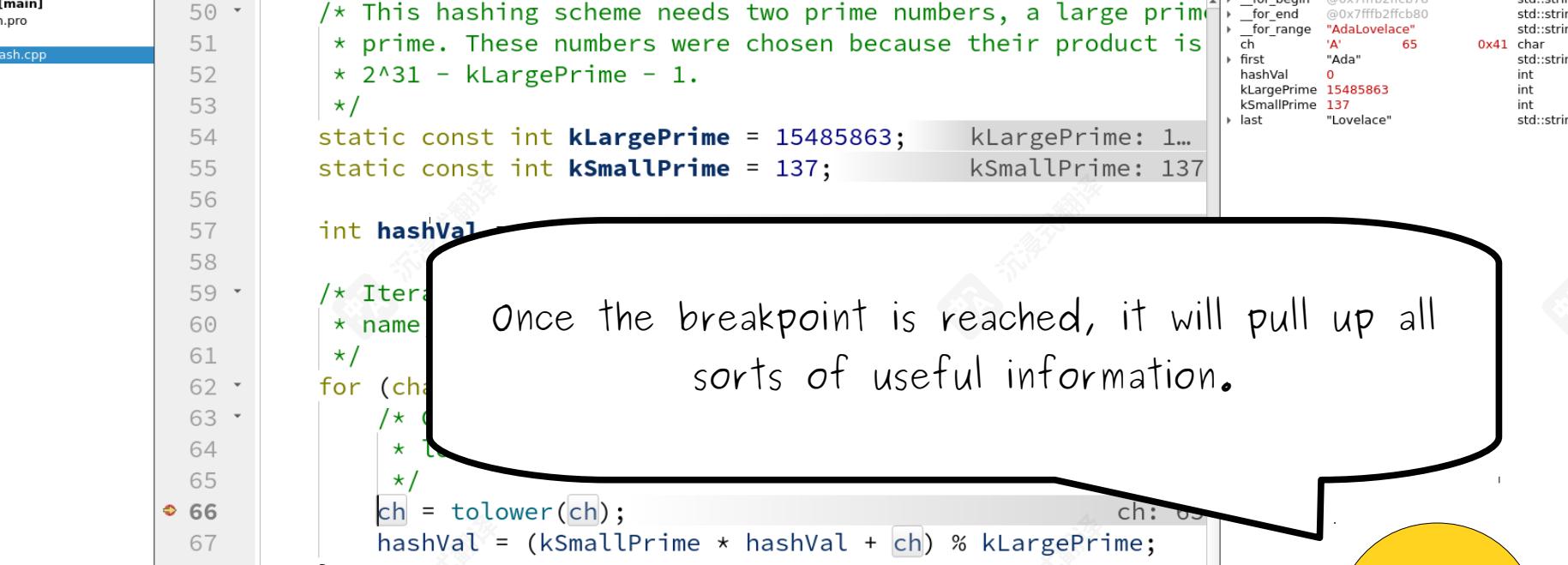
7 clone3 clone3.S 81 0x7ffff6126a00 7 clone3()

Type to locate (Ctrl) Issues Search Results Application Output Compile Output QML Debugger Console General Messages Version Control Test Results

断点以便我们可以
周围，点击你想要暂



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



The screenshot shows the Qt Creator interface with a debugger window open. The code being debugged is NameHash.cpp, specifically the function nameHash(string, string) -> int. A breakpoint is set at line 66, where the variable 'ch' has a value of 65. The debugger pane on the right displays the current state of variables:

Name	Value	Type
_for_begin	@0x7ffb2ffcb78	std::string::iterator
_for_end	@0x7ffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

0x41

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



一旦达到断点，它将拉起所有
各种有用的信息。

```
File Edit View Build Debug Analyze Tools Window Help
NameHash.cpp @ NameHash [main] - Qt Creator
Projects NameHash [main]
Sources NameHash.cpp
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
Line: 66, Col: 9
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;    kLargePrime: 1...
static const int kSmallPrime = 137;          kSmallPrime: 137

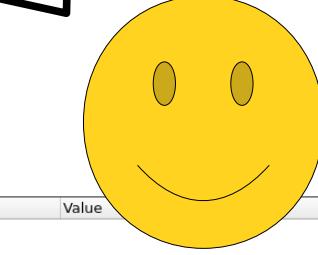
int hashVal

/* Iterat
 * name
 */
for (char ch : name)
{
    /* C
     * to
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

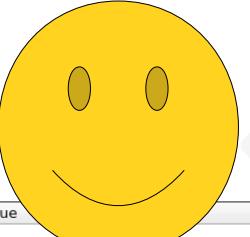
return hashVal;

NameValueType
__for_begin@0x7fff82ffcb78std::string::iter.
__for_end@0x7fff82ffcb80std::string::iter.
__for_range"AdaLovelace"std::string &
ch'A'65char
first"Ada"std::string
hashVal0int
kLargePrime15485863int
kSmallPrime137int
last"Lovelace"std::string
0x41
```

达到断点，它将拉起所
各种有用的信



The yellow arrow points out where we are right now.



Name	Type	Value
_for_begin	std::string::iterator	@0x7ffb2ffcb78
_for_end	std::string::iterator	@0x7ffb2ffcb80
_for_range	std::string &	"AdaLovelace"
ch	char	'A'
first	int	65
hashVal	int	0
kLargePrime	int	15485863
kSmallPrime	int	137
last	std::string	"Lovelace"

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137

56
57 int hashVal
58
59 /* Iterates over the string
60 * name
61 */
62 for (char ch : name)
63 /* Converts character to lowercase
64 * before adding to hashVal
65 */
66 ch = tolower(ch); ch: 65
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69
70 return hashVal;
71
72 }

Yellow arrow points to line 66.

Yellow smiley face icon.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
+ 1	nameHash	NameHash.cpp	66	0x555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::_Function_ha...			0x5555556037fc								
4	GThreadStd::run()			0x5555555e6616								
5	??			0x7ffff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 OML Debugger Console 6 General Messages 7 Version Control 8 Test Results

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

```
/* This hashing scheme needs two prime numbers, a large prime.
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;    kLargePrime: 1...
static const int kSmallPrime = 137;          kSmallPrime: 137

int hashVal

/* Iterates over the string 'name'.
 * name
 */
for (char ch : name)
    /* Converts character to lowercase.
     * tolower
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

}

return hashVal;
```

Name	Type	Value
_for_begin	std::string::iterator	0x7ffb2ffcb78
_for_end	std::string::iterator	0x7ffb2ffcb80
_for_range	std::string::const_iterator	"AdaLovelace"
ch	char	'A'
first	std::string	"Ada"
hashVal	int	0
kLargePrime	int	15485863
kSmallPrime	int	137
last	std::string	"Lovelace"

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)	
2	studentMain	NameHash.cpp	31	0x5555556037fc							
3	std::function<_T>::operator()			0x5555555e6616							
4	GThreadStd::run()			0x7fffff64dc2b3							
5	??			0x7fffff64dc2b3							
6	start_thread	pthread_create.c	442	0x7fffff094b43							
7	clone3	clone3.S	81	0x7fffff6126a00							

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Projects Help

NameHash.cpp @ NameHash [main] - Qt Creator

Projects NameHash [main] Sources NameHash.cpp

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137

56
57 int hashVal
58
59 /* Iterates over the string.
60 * name
61 *
62 * for (char ch : name)
63 * {
64 * /* Convert character to lowercase.
65 * * To do this, we subtract the ASCII value of 'A' from the character's value.
66 * *
67 * ch = tolower(ch);
68 * hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69 * }
70
71 return hashVal;

72 }

Line: 66, Col: 9

Name Value Type
__for_begin @0x7fffb2ffcb78 std::string::iterator
__for_end @0x7fffb2ffcb80 std::string::iterator
__for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

0x41

调用栈显示我们如何进入当前函数。

ch: 65

Debugger GDB for "NameHash" Threads #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address
1 nameHash NameHash.cpp 66 0x7fffb2ff555ab2d3
2 studentMain NameHash.cpp 31 0x7fffb2ff555ab0fb
3 std::Function_han...
4 GThreadStd::run()
5 ??
6 start_thread pthread_create.c 442 0x7ffff6094b43
7 clone3 clone3.S 81 0x7ffff6126a00

Number Function File Line Address Con Ignore Threads
1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

Views

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Debug Projects Help

NameHash.cpp @ NameHash [main] - Qt Creator

Projects NameHash [main] Sources NameHash.cpp

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137

56
57 int hashVal
58
59 /* Iterates over each character in the name
60 * name
61 */
62 for (char ch : name)
63 /* Converts character to lowercase
64 * tolower
65 */
66 ch = tolower(ch); ch: 65
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69
70 return hashVal;

Line: 66, Col: 9

Name Value Type
_for_begin @0x7ffb2ffcb78 std::string::iterator
_for_end @0x7ffb2ffcb80 std::string::iterator
_for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

0x41

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

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



现在，让我们看看如何读取这个函数中的
变量的值。

```
File Edit View Build Debug Analyze Tools Window Help
NameHash.cpp @ NameHash [main] - Qt Creator
Projects Projects NameHash.cpp Line: 66, Col: 9
NameHash [main]
  NameHash.pro
  Sources
    NameHash.cpp
 50  /* This hashing scheme needs two prime numbers, a large prime
 51  * prime. These numbers were chosen because their product is
 52  * 2^31 - kLargePrime - 1.
 53  */
 54  static const int kLargePrime = 15485863;      kLargePrime: 1...
 55  static const int kSmallPrime = 137;           kSmallPrime: 137
 56
 57  int hashVal
 58
 59  /* Iterates over each character in the name
 60  * name
 61  */
 62  for (char ch : name)
 63  {
 64  /* Converts character to lowercase
 65  * tolower(ch)
 66  */
 67  ch = tolower(ch);
 68  hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
 69
 70
 71  return hashVal;
 72 }
```

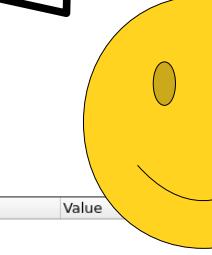
现在，让我们看看如何读取这个函数中的
变量的值。

ch: 65

Name	Value	Type
__for_begin	@0x7fff82ffcb78	std::string::iterator
__for_end	@0x7fff82ffcb80	std::string::iterator
__for_range	"AdaLovelace"	std::string &
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

0x41

让我们看看如何读取这个
变量的值。



NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

Look up at this panel over here.

large prime product is

large prime product is

0x41

Name	Value	Type
__for_begin	@0x7fff2ffcb78	std::string::iterator
__for_end	@0x7fff2ffcb80	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

0x41

Name	Value	Type
__for_begin	@0x7fff2ffcb78	std::string::iterator
__for_end	@0x7fff2ffcb80	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

hashVal: 0

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */

for (char ch: first + last) { first: "Ada" last: "Love..."

/* Convert the input character to lower case. The numeric * lower-case letters are always less than 127. */

ch = tolower(ch);

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

return hashVal;

hashVal: 0

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */

for (char ch: first + last) { first: "Ada" last: "Love..."

/* Convert the input character to lower case. The numeric * lower-case letters are always less than 127. */

ch = tolower(ch);

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

return hashVal;

Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

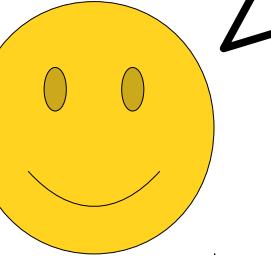
Debugger GDB for "NameHash" File Line Address Number Function File Line Address Con Ignore Threads

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x555555ab0fb								
3	std::Function_h...			0x5555556037fc								
4	GThreadStd::run()			0x555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 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 now.
(Don't worry if you see different values or "not accessible" on your system - that's okay!)



Prime: 1...
kSmallPrime: 137

```
int hashVal = 0;                                hashVal: 0
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) {      first: "Ada"    last: "Love...
/* Convert the input character to lower case. The numeric
 * lower-case letters are always less than 127.
 */
ch = tolower(ch);                  ch: 65
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;
```

large prime product is

Name	Value	Type
__for_begin	@0x7fffb2ffcb78	std::string::iterator
__for_end	@0x7fffb2ffcb80	std::string::iterator
ch	"AdaLovelace"	std::string &
first	'A'	char
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

File Edit View Build
Projects NameHash
Welcome Edit Design Debug Projects Help
File Edit View Build
Projects NameHash
Welcome Edit Design Debug Projects Help
File Edit View Build
Projects NameHash
Welcome Edit Design Debug Projects Help

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x555555ab0fb	2	studentMain	NameHash.cpp	31	0x555555ab0fb			
3	std::Function_h...			0x5555556037fc	3	std::Function_h...			0x5555556037fc			
4	GThreadStd::run()			0x555555e616	4	GThreadStd::run()			0x555555e616			
5	??			0x7ffff64dc2b3	5	??			0x7ffff64dc2b3			
6	start_thread	pthread_create.c	442	0x7ffff6094b43	6	start_thread	pthread_create.c	442	0x7ffff6094b43			
7	clone3	clone3.S	81	0x7ffff6126a00	7	clone3	clone3.S	81	0x7ffff6126a00			

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

这个窗口让你可以查看所有值
当前在范围内的局部变量。
(如果看到不同的值或“不可访问”在你的系统上——那没关系！)
您系统上的“accessible”——那没关系！



Prime: 1...
kSmallPrime: 137

```
int hashVal = 0;                                hashVal: 0
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) {      first: "Ada"    last: "Love...
/* Convert the input character to lower case. The numeric
 * lower-case letters are always less than 127.
 */
ch = tolower(ch);                  ch: 65
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;
```

large prime product is

Name	Value	Type
__for_begin	@0x7fffb2ffcb78	std::string::iterator
__for_end	@0x7fffb2ffcb80	std::string::iterator
ch	"AdaLovelace"	std::string &
first	'A'	char
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

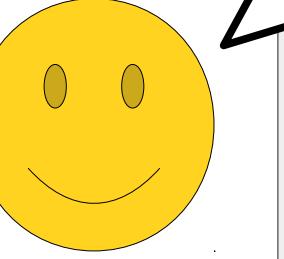
File Edit View Build
Projects NameHash
Welcome Edit Design Debug Projects Help
File Edit View Build
Projects NameHash
Welcome Edit Design Debug Projects Help
File Edit View Build
Projects NameHash
Welcome Edit Design Debug Projects Help

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x555555ab0fb	2	studentMain	NameHash.cpp	31	0x555555ab0fb			
3	std::Function_h...			0x5555556037fc	3	std::Function_h...			0x5555556037fc			
4	GThreadStd::run()			0x555555e616	4	GThreadStd::run()			0x555555e616			
5	??			0x7ffff64dc2b3	5	??			0x7ffff64dc2b3			
6	start_thread	pthread_create.c	442	0x7ffff6094b43	6	start_thread	pthread_create.c	442	0x7ffff6094b43			
7	clone3	clone3.S	81	0x7ffff6126a00	7	clone3	clone3.S	81	0x7ffff6126a00			

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

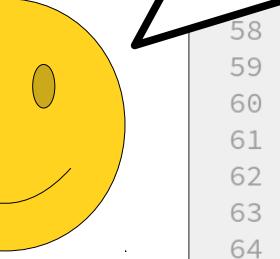
Depending on what OS you're using, these might be in a different order, and there might be some weird-looking ones in there in addition to nicer ones like ch and hashVal.



```
File Edit View Build  
Projects NameHash  
Welcome Edit Design Debug Projects ? Help  
NameHash Source NameHash  
Line: 66, Col: 9  
large prime product is  
Prime: 1... kSmallPrime: 137  
  
int hashVal = 0; hashVal: 0  
58  
59 /* Iterate across all the characters in the first name, then  
60 * name, updating the hash at each step.  
61 */  
62 for (char ch: first + last) { first: "Ada" last: "Love... ch: 65  
63 /* Convert the input character to lower case. The numeric  
64 * lower-case letters are always less than 127.  
65 */  
66 ch = tolower(ch); ch: 65  
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
68 }  
69  
70 return hashVal;  
71 }  
72 }
```

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.
Level Function File Line Address Number Function File Line Address Con Ignore Threads
1 nameHash NameHash.cpp 66 0x555555ab2d3 1 nameHash(std::string, std::string)
2 studentMain NameHash.cpp 31 0x555555ab0fb ...eHash.cpp 66 ...ab2d3 (all)
3 std::Function_h... 0x5555556037fc
4 GThreadStd::run() 0x555555e616
5 ?? 0xfffff64dc2b3
6 start_thread pthread_create.c 442 0xfffff6094b43
7 clone3 clone3.S 81 0xfffff6126a00

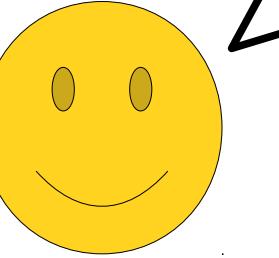
根据您使用的操作系统，这些可能会以不同的顺序排列，并且可能会有一些看起来奇怪的项，除了更美观的项之外，比如 ch 和 hashVal。



```
File Edit View Build  
Projects NameHash  
Welcome Edit Design Debug Projects ? Help  
NameHash Source NameHash  
Line: 66, Col: 9  
large prime product is  
Prime: 1... kSmallPrime: 137  
  
int hashVal = 0; hashVal: 0  
58  
59 /* Iterate across all the characters in the first name, then  
60 * name, updating the hash at each step.  
61 */  
62 for (char ch: first + last) { first: "Ada" last: "Love... ch: 65  
63 /* Convert the input character to lower case. The numeric  
64 * lower-case letters are always less than 127.  
65 */  
66 ch = tolower(ch); ch: 65  
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
68 }  
69  
70 return hashVal;  
71 }  
72 }
```

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.
Level Function File Line Address Number Function File Line Address Con Ignore Threads
1 nameHash NameHash.cpp 66 0x555555ab2d3 1 nameHash(std::string, std::string)
2 studentMain NameHash.cpp 31 0x555555ab0fb ...eHash.cpp 66 ...ab2d3 (all)
3 std::Function_h... 0x5555556037fc
4 GThreadStd::run() 0x555555e616
5 ?? 0xfffff64dc2b3
6 start_thread pthread_create.c 442 0xfffff6094b43
7 clone3 clone3.S 81 0xfffff6126a00

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



large prime product is

Prime: 1... kSmallPrime: 137

```
int hashVal = 0; hashVal: 0
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
/* Convert the input character to lower case. The numeric
 * lower-case letters are always less than 127.
 */
ch = tolower(ch); ch: 65
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;
```

Line: 66, Col: 9

Name	Value	Type
__for_begin	@0x7fffb2ffcb78	std::string::iterator
__for_end	@0x7fffb2ffcb80	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

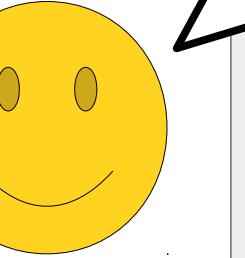
File Edit View Build Projects Welcome Edit Design Debug Projects Help

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15. Level Function File Line Address Number Function File Line Address Con Ignore Threads 1 nameHash NameHash.cpp 66 0x5555555ab2d3 1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

2 studentMain NameHash.cpp 31 0x5555555ab0fb 2 studentMain NameHash.cpp 31 0x5555555ab0fb 3 std::Function_h... 0x55555556037fc 3 std::Function_h... 0x55555556037fc 4 GThreadStd::run() 0x5555555e616 4 GThreadStd::run() 0x5555555e616 5 ?? 0x7ffff64dc2b3 5 ?? 0x7ffff64dc2b3 6 start_thread pthread_create.c 442 0x7ffff6094b43 6 start_thread pthread_create.c 442 0x7ffff6094b43 7 clone3 clone3.S 81 0x7ffff6126a00 7 clone3 clone3.S 81 0x7ffff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

如果我们忽略那些看起来奇怪的，我们就可以看到一些漂亮、熟悉的名字。



large prime product is

Prime: 1... kSmallPrime: 137

```
int hashVal = 0; hashVal: 0
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
/* Convert the input character to lower case. The numeric
 * lower-case letters are always less than 127.
 */
ch = tolower(ch); ch: 65
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;
```

Line: 66, Col: 9

Name	Value	Type
__for_begin	@0x7fffb2ffcb78	std::string::iterator
__for_end	@0x7fffb2ffcb80	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

File Edit View Build Projects Welcome Edit Design Debug Projects Help

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15. Level Function File Line Address Number Function File Line Address Con Ignore Threads 1 nameHash NameHash.cpp 66 0x5555555ab2d3 1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

2 studentMain NameHash.cpp 31 0x5555555ab0fb 2 studentMain NameHash.cpp 31 0x5555555ab0fb 3 std::Function_h... 0x55555556037fc 3 std::Function_h... 0x55555556037fc 4 GThreadStd::run() 0x5555555e616 4 GThreadStd::run() 0x5555555e616 5 ?? 0x7ffff64dc2b3 5 ?? 0x7ffff64dc2b3 6 start_thread pthread_create.c 442 0x7ffff6094b43 6 start_thread pthread_create.c 442 0x7ffff6094b43 7 clone3 clone3.S 81 0x7ffff6126a00 7 clone3 clone3.S 81 0x7ffff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60

```
/* This hashing scheme needs two prime numbers, a large prime.  
 * These numbers were chosen because their product is  
 * 2^31 - kLargePrime - 1.  
 */  
static const int kLargePrime = 15485863;  
static const int kSmallPrime = 137;
```

int hashVal = 0;

```
/* Iterate across all the characters in the first name, then  
 * name, updating the hash at each step.
```

0x41 Name Value Type
for_begin @0x7fffb2ffcb78 std::string::iterator
for_end @0x7fffb2ffcb80 std::string::iterator
for_range "AdaLovelace" std::string &
ch 'A' 65 char
first 0 std::string int
hashVal 15485863 int
kLargePrime 137 int
kSmallPrime 137 int
last "Lovelace" std::string

50
51
52
53
54
55
56
57
58
59
60

```
/* This hashing scheme needs two prime numbers, a large prime.  
 * These numbers were chosen because their product is  
 * 2^31 - kLargePrime - 1.  
 */  
static const int kLargePrime = 15485863;  
static const int kSmallPrime = 137;
```

int hashVal = 0;

```
/* Iterate across all the characters in the first name, then  
 * name, updating the hash at each step.
```

0x41 Name Value Type
for_begin @0x7fffb2ffcb78 std::string::iterator
for_end @0x7fffb2ffcb80 std::string::iterator
for_range "AdaLovelace" std::string &
ch 'A' 65 char
first 0 std::string int
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

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

例如，这里你可以看到 kLargePrime 和 kSmallPrime 的值，它们与它们被声明时的值相匹配。

return hashVal;

72

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1	nameHash	NameHash.cpp	66	0x555555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)
2	studentMain	NameHash.cpp	31	0x555555ab0fb	3	std::Function_h...	0x5555556037fc			
3	std::Function_h...				4	GThreadStd::run()	0x5555556037fc			
4	GThreadStd::run()				5	??	0x7fff64dc2b3			
5	??				6	start_thread	0x7fff6094b43			
6	start_thread	pthread_create.c	442	0x7fff6094b43	7	clone3	pthread_create.c	442	0x7fff6126a00	
7	clone3	clone3.S	81	0x7fff6126a00						

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 66, Col: 9

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 66, Col: 9

50
51
52
53
54
55
56
57
58
59
60

/* This hashing scheme needs two prime numbers, a large prime.
 * prime. These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.
 */

static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137, kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.

"Love..."
numerical
ch: 65
time;

0x41

Name Value Type
_for_begin @0x7fffb2ffcb78 std::string::iter...
_for_end @0x7fffb2ffcb80 std::string::iter...
_for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

50
51
52
53
54
55
56
57
58
59
60

/* This hashing scheme needs two prime numbers, a large prime.
 * prime. These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.
 */

static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137, kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.

"Love..."
numerical
ch: 65
time;

0x41

Name Value Type
_for_begin @0x7fffb2ffcb78 std::string::iter...
_for_end @0x7fffb2ffcb80 std::string::iter...
_for_range "AdaLovelace" std::string &
ch 'A' 65 char
first "Ada" std::string
hashVal 0 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

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

我们也可以看到, 在这个点上, hashVal 仍然是零。

return hashVal;

72

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 66 0x555555ab2d3 1 nameHash(std::string, std::string)

2 studentMain NameHash.cpp 31 0x555555ab0fb 2 studentMain

3 std::Function_h... 0x5555556037fc 3 std::Function_h...

4 GThreadStd::run() 0x555555e616 4 GThreadStd::run()

5 ?? 0x7ffff64dc2b3 5 ??

6 start_thread pthread_create.c 442 0x7ffff6094b43 6 start_thread

7 clone3 clone3.S 81 0x7ffff6126a00 7 clone3

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

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

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;    kLargePrime: 1...
static const int kSmallPrime = 137;          kSmallPrime: 137

int hashVal = 0;                           hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
```

A large black callout bubble originates from the bottom left, pointing towards the code. Inside the bubble, there is Chinese text:

我们一步步地走遍程序，
我们会看到这些值的变化。

At the bottom left, there is a large yellow smiley face icon. A black arrow points from the smiley face towards the code editor area.

The status bar at the bottom shows various tabs: Issues, Search Results, Application Output, Compile Output, QML Debugger Console, General Messages, Version Control, Test Results, and a search bar.

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

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

large prime product is

Prime: 1...
Prime: 137

hashVal = 0;

/* Iterate across all the characters in the first name, then * name, updating the hash at each step.

*/

for (char ch: first + last) { first: "Ada" last: "Lovelace"
/* Convert the input character to lower case. The numeric
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;

large prime product is

Prime: 1...
Prime: 137

hashVal = 0;

/* Iterate across all the characters in the first name, then * name, updating the hash at each step.

*/

for (char ch: first + last) { first: "Ada" last: "Lovelace"
/* Convert the input character to lower case. The numeric
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;

File Edit View Build Debug Analyze

File Edit View Build Debug Analyze

Projects Projects

Welcome Welcome

Edit Edit

Design Design

Debug Debug

Projects Projects

Welcome Welcome

Edit Edit

Design Design

Debug Debug

Help Help

Help Help

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 66 0x555555ab2d3 1 nameHash(std::string, std::string)

2 studentMain NameHash.cpp 31 0x555555ab0fb ...eHash.cpp 66 ...ab2d3 (all)

3 std::Function_h... 0x55555556037fc

4 GThreadStd::run() 0x5555555e616

5 ?? 0x7fff64dc2b3

6 start_thread pthread_create.c 442 0x7fff6094b43

7 clone3 clone3.S 81 0x7fff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

File Edit View Build Debug Analyze

File Edit View Build Debug Analyze

Projects Projects

Welcome Welcome

Edit Edit

Design Design

Debug Debug

Projects Projects

Welcome Welcome

Edit Edit

Design Design

Debug Debug

Help Help

Help Help

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 66 0x555555ab2d3 1 nameHash(std::string, std::string)

2 studentMain NameHash.cpp 31 0x555555ab0fb ...eHash.cpp 66 ...ab2d3 (all)

3 std::Function_h... 0x55555556037fc

4 GThreadStd::run() 0x5555555e616

5 ?? 0x7fff64dc2b3

6 start_thread pthread_create.c 442 0x7fff6094b43

7 clone3 clone3.S 81 0x7fff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 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."

hashVal = 0;

```
/* Iterate across all the characters in the first name, then * name, updating the hash at each step.
```

for (char ch: first + last) { first: "Ada" last: "Lovelace" /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127. */ ch = tolower(ch); hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; }

return hashVal;

large prime product is

Name	Value	Type
__for_begin	@0x7fffb2ffcb78	std::string::iterator
__for_end	@0x7fffb2ffcb80	std::string::iterator
__for_range	"AdaLovelace"	std::string &
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

hashVal = 0;

```
/* Iterate across all the characters in the first name, then * name, updating the hash at each step.
```

for (char ch: first + last) { first: "Ada" last: "Lovelace" /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127. */ ch = tolower(ch); hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; }

return hashVal;

large prime product is

Name	Value	Type
__for_begin	@0x7fffb2ffcb78	std::string::iterator
__for_end	@0x7fffb2ffcb80	std::string::iterator
__for_range	"AdaLovelace"	std::string &
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze

Projects

Welcome

Edit

Design

Debug

Projects

Help

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

large prime product is

Prime: 1...

Prime: 137

hashVal = 0;

hashVal: 0

/* Iterate across all the characters in the first name, then * name, updating the hash at each step.

*/

for (char ch: first + last) { first: "Ada" last: "Lovelace" }

/* Convert the input character to lower case. The numeric * lower-case letters are always less than 127.

*/

ch = tolower(ch);

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

return hashVal;

Line: 66, Col: 9

Name Value Type

_for_begin @0x7fffb2ffcb78 std::string::iterator

_for_end @0x7fffb2ffcb80 std::string::iterator

_for_range "AdaLovelace" std::string &

ch 'A' 65 char

first "Ada" 0 std::string int

hashVal 15485863 std::string int

kLargePrime 137 std::string int

last "Lovelace" std::string

0x41

Name HashValue Type

_for_begin @0x7fffb2ffcb78 std::string::iterator

_for_end @0x7fffb2ffcb80 std::string::iterator

_for_range "AdaLovelace" std::string &

ch 'A' 65 char

first "Ada" 0 std::string int

hashVal 15485863 std::string int

kLargePrime 137 std::string int

last "Lovelace" std::string

0x41

Remember (从很久以前起) 我们输入了名字 Ada Lovelace?

large prime product is

Prime: 1...

Prime: 137

hashVal = 0;

hashVal: 0

/* Iterate across all the characters in the first name, then * name, updating the hash at each step.

*/

for (char ch: first + last) { first: "Ada" last: "Lovelace" }

/* Convert the input character to lower case. The numeric * lower-case letters are always less than 127.

*/

ch = tolower(ch);

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

return hashVal;

Line: 66, Col: 9

Name Value Type

_for_begin @0x7fffb2ffcb78 std::string::iterator

_for_end @0x7fffb2ffcb80 std::string::iterator

_for_range "AdaLovelace" std::string &

ch 'A' 65 char

first "Ada" 0 std::string int

hashVal 15485863 std::string int

kLargePrime 137 std::string int

last "Lovelace" std::string

0x41

Name HashValue Type

_for_begin @0x7fffb2ffcb78 std::string::iterator

_for_end @0x7fffb2ffcb80 std::string::iterator

_for_range "AdaLovelace" std::string &

ch 'A' 65 char

first "Ada" 0 std::string int

hashVal 15485863 std::string int

kLargePrime 137 std::string int

last "Lovelace" std::string

0x41

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 66 0x555555ab2d3 1 nameHash(std::string, std::string)

2 studentMain NameHash.cpp 31 0x555555ab0fb ...eHash.cpp 66 ...ab2d3 (all)

3 std::Function_h... 0x5555556037fc

4 GThreadStd::run() 0x555555e616

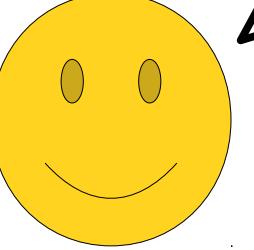
5 ?? 0x7fff64dc2b3

6 start_thread pthread_create.c 442 0x7fff6094b43

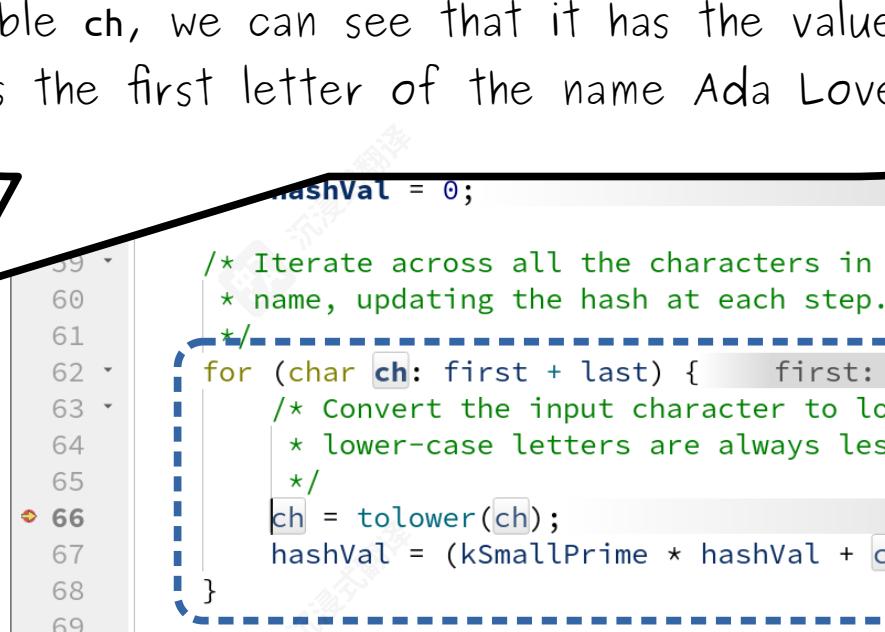
7 clone3 clone3.S 81 0x7fff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

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.



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.



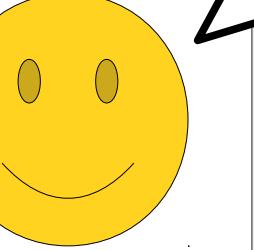
```
analyze

    hashVal = 0;

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

    return hashVal;
}
```

如果我们看一下当前变量的值，
我们可以看到它有值 A。
那是Ada Lovelace的名字的第一个字母。



如果我们看一下当前变量的值，
我们可以看到它有值 A。
那是Ada Lovelace的名字的第一个字母。

```
analyze

Line: 66, Col: 9
large prime
product is
Prime: 1...
Prime: 137
hashVal: 0

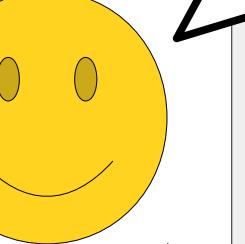
Name Value Type
__for_begin @0x7fff2ffcb78 std::string::iterator
__for_end @0x7fff2ffcb80 std::string::iterator
for_range "AdaLovelace" std::string &&
ch 'A' 65 char
kSmallPrime 137 std::string int
kLargePrime 1548563 int
last "Lovelace" std::string int
0x41

hashVal = 0;
/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) {    first: "Ada"    last: "Love...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
```

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.



我们知道我们在哪里（第66行），以及我们如何
主调用**nameHash**），以及此时程序中的值在
这个点。

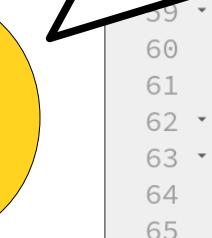
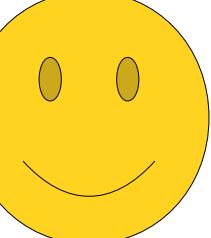


NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

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!

现在, 让我们做点真正酷的事情——我们将逐行运行这个程序, 观察每一步发生什么在每个步骤中都会发生!



hashVal = 0;

/* Iterate across all the characters in the first name, then name, updating the hash at each step.

*/

```
for (char ch: first + last) {    first: "Ada"    last: "Love...  
    /* Convert the input character to lower case. The numeric  
     * lower-case letters are always less than 127.  
     */  
    ch = tolower(ch);  
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
}
```

return hashVal;

Line: 66, Col: 9

Name	Value	Type
_for_begin	@0x7ffb2ffcb78	std::string::iterator
_for_end	@0x7ffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

Line: 66, Col: 9

Name	Value	Type
_for_begin	@0x7ffb2ffcb78	std::string::iterator
_for_end	@0x7ffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

File Edit View Build Debug Analyze Projects Welcome Edit Design Debug Projects Help

File Edit View Build Debug Analyze Projects Welcome Edit Design Debug Projects Help

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
    */
    ch =
has
return
```

Name Value Type

- _for_begin @0x7ffb2ffcb78 std::string::iter...
- _for_end @0x7ffb2ffcb80 std::string::iter...
- _for_range "AdaLovelace" std::string &
- ch 'A' 65 0x41 char
- first "Ada" std::string
- hashVal 0 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
    */
    ch =
has
return
```

Name Value Type

- _for_begin @0x7ffb2ffcb78 std::string::iter...
- _for_end @0x7ffb2ffcb80 std::string::iter...
- _for_range "AdaLovelace" std::string &
- ch 'A' 65 0x41 char
- first "Ada" std::string
- hashVal 0 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

右上方有三个图标，分别代表不同的功能：停止、步进和单步。

右上方有三个图标，分别代表不同的功能：停止、步进和单步。

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

堆栈跟踪的上方，你会看到那里有一些小的按钮图标。

下方是调试器窗口，显示了调用堆栈、线程信息和断点状态。

下方是调试器窗口，显示了调用堆栈、线程信息和断点状态。

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
    */
    ch =
hashVal += ch;

return hashVal;
}
```

Name Value Type

- _for_begin @0x7ffb2ffcb78 std::string::iter...
- _for_end @0x7ffb2ffcb80 std::string::iter...
- _for_range "AdaLovelace" std::string &
- ch 'A' 65 0x41 char
- first "Ada" std::string
- hashVal 0 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
    */
    ch =
hashVal += ch;

return hashVal;
}
```

Name Value Type

- _for_begin @0x7ffb2ffcb78 std::string::iter...
- _for_end @0x7ffb2ffcb80 std::string::iter...
- _for_range "AdaLovelace" std::string &
- ch 'A' 65 0x41 char
- first "Ada" std::string
- hashVal 0 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

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

这些按钮允许您继续程序，停止程序，逐行执行等。

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15. Views

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15. Views

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
    */
    ch =
hashVal =
}
```

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."

Projects NameHash [main] Sources NameHash.cpp

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
    */
    ch =
hashVal =
}
```

移动鼠标，使其悬停在左侧第三个按钮上。如果你悬停在其上，它应该显示“跳过。”

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
    */
    ch:
    hashVal += ch;
}
return hashVal;
```

Name Value Type

- _for_begin @0x7ffb2ffcb78 std::string::iter...
- _for_end @0x7ffb2ffcb80 std::string::iter...
- _for_range "AdaLovelace" std::string &
- ch 'A' 65 char
- first "Ada" std::string
- hashVal 0 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

0x41

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 0

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { first: "Ada" last: "Love...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
    */
    ch:
    hashVal += ch;
}
return hashVal;
```

Name Value Type

- _for_begin @0x7ffb2ffcb78 std::string::iter...
- _for_end @0x7ffb2ffcb80 std::string::iter...
- _for_range "AdaLovelace" std::string &
- ch 'A' 65 char
- first "Ada" std::string
- hashVal 0 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

0x41

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...

一旦您确定您位于“单步跳过”按钮 - 并且不是“单步进入”或“单步跳出”按钮 - 然后点击它! 当您这样做时...

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.  
 * These numbers were chosen because their product is  
 * 2^31 - kLargePrime - 1.  
 */  
  
static const int kLargePrime = 15485863;  
static const int kSmallPrime = 137;  
  
int hashVal = 0;  
  
/* Iterate a  
 * name, upda  
 */  
for (char ch : name)  
    /* Convert  
     * lower  
     */  
    ch = tolower(ch);  
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
  
return hashVal;
```

...your window should look something like this.

Name Value Type
_for_begin @0x7ffffb2ffcc
_for_end @0x7ffffb2ffcc
_for_range "AdaLovelac
ch 'a'
first "Ada"
hashVal 0
kLargePrime 15485863
kSmallPrime 137
last "Lovelace"

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.  
 * These numbers were chosen because their product is  
 * 2^31 - kLargePrime - 1.  
 */  
  
static const int kLargePrime = 15485863;  
static const int kSmallPrime = 137;  
  
int hashVal = 0;  
  
/* Iterate a  
 * name, upda  
 */  
for (char ch : name)  
    /* Convert  
     * lower  
     */  
    ch = tolower(ch);  
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
  
return hashVal;
```

...你的窗口应该看起来像这样。

Name Value Type
_for_begin @0x7ffffb2ffcc
_for_end @0x7ffffb2ffcc
_for_range "AdaLovelac
ch 'a'
first "Ada"
hashVal 0
kLargePrime 15485863
kSmallPrime 137
last "Lovelace"

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".
Level Function File Line Address Number Function File Line Address Con Ignore Threads
1 nameHash NameHash.cpp 67 0x5555555ab2e1 1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)
2 studentMain NameHash.cpp 31 0x5555555ab0fb 3 std::Function_h... 0x55555556037fc
3 std::Function_h... 0x55555556037fc 4 GThreadStd::run() 0x5555555e616
4 GThreadStd::run() 0x5555555e616 5 ?? 0xfffff64dc2b3
6 start_thread pthread_create.c 442 0x7ffff6094b43 6 start_thread pthread_create.c 442 0x7ffff6094b43
7 clone3 clone3.S 81 0x7ffff6126a00 7 clone3 clone3.S 81 0x7ffff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.  
 * These numbers were chosen because their product is  
 * 2^31 - kLargePrime - 1.  
 */  
  
static const int kLargePrime = 15485863;  
static const int kSmallPrime = 137;  
  
int hashVal = 0;  
  
/* Iterate a  
 * name, upda  
 */  
for (char ch : name)  
    /* Convert  
     * lower  
     */  
    ch = tolower(ch);  
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
  
return hashVal;
```

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.  
 * These numbers were chosen because their product is  
 * 2^31 - kLargePrime - 1.  
 */  
  
static const int kLargePrime = 15485863;  
static const int kSmallPrime = 137;  
  
int hashVal = 0;  
  
/* Iterate a  
 * name, upda  
 */  
for (char ch : name)  
    /* Convert  
     * lower  
     */  
    ch = tolower(ch);  
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
  
return hashVal;
```

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

Threads: #15 NameHash Stopped: "end-stepping-range".

Threads: #15 NameHash Stopped: "end-stepping-range".

Views

Views

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x5555555ab2e1	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::function<_H>::_Function_ha...			0x55555556037fc								
4	GThreadStd::run()			0x5555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Smiley face icon

Smiley face icon

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

好的！有些事情变了。让我们看看发生了什么。

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.
 */

static const int kLargePrime = 15485863;
static const int kSmallPrime = 137;

int hashVal = 0;

/* Iterate a name, update hashVal
 * name, update hashVal
 */

for (char ch : name)
 /* Convert character to lower case
 * lower
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

}

return hashVal;

Line: 67, Col: 9

Name Value Type
_for_begin @0x7fffb2ffca
_for_end @0x7fffb2ffca
_for_range "AdaLovelac
ch 'a'
first hashVal 0
kLargePrime 15485863
kSmallPrime 137
last "Lovelace"

Line: 67, Col: 9

Name Value Type
_for_begin @0x7fffb2ffca
_for_end @0x7fffb2ffca
_for_range "AdaLovelac
ch 'a'
first hashVal 0
kLargePrime 15485863
kSmallPrime 137
last "Lovelace"

Line: 67, Col: 9

Name Value Type
_for_begin @0x7fffb2ffca
_for_end @0x7fffb2ffca
_for_range "AdaLovelac
ch 'a'
first hashVal 0
kLargePrime 15485863
kSmallPrime 137
last "Lovelace"

Line: 67, Col: 9

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.
 */

static const int kLargePrime = 15485863;
static const int kSmallPrime = 137;

int hashVal = 0;

/* Iterate a name, update hashVal
 * name, update hashVal
 */

for (char ch : name)
 /* Convert character to lower case
 * lower
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

}

return hashVal;

Line: 67, Col: 9

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

首先,请注意我们友好的黄色箭头朋友
现在指向第67行。

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 67 0x555555ab2e1
2 studentMain NameHash.cpp 31 0x555555ab0fb
3 std::Function_h... 0x5555556037fc
4 GThreadStd::run() 0x555555e616
5 ?? 0x7fff64dc2b3
6 start_thread pthread_create.c 442 0x7fff6094b43
7 clone3 clone3.S 81 0x7fff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;
static const int kSmallPrime = 137;

int hashVal = 0;

/* Iterate a
 * name, upda
 */
for (char ch
    /* Conv
     * lower
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

}

return hashVal;
```

Line: 67, Col: 9

Name Value Type

Name	Value	Type
_for_begin	@0x7ffffb2ff0	
_for_end	@0x7ffffb2ff0	
_for_range	"AdaLovelac	
ch	'a'	
first	"Ada"	
hashVal	0	
kLargePrime	15485863	
kSmallPrime	137	
last	"Lovelace"	

Line: 67, Col: 9

Name Value Type

Name	Value	Type
_for_begin	@0x7ffffb2ff0	
_for_end	@0x7ffffb2ff0	
_for_range	"AdaLovelac	
ch	'a'	
first	"Ada"	
hashVal	0	
kLargePrime	15485863	
kSmallPrime	137	
last	"Lovelace"	

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

我们现在位于停下的那一行后面的行 我们刚刚运行了程序的一行! 很酷!

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Debug Projects Help

NameHash.cpp @ NameHash [main] - Qt Creator

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

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863;
55 static const int kSmallPrime = 137;
56
57 int hashVal = 0;
58
59 /* Iterate a
60 * name, update
61 * hashVal.
62 */
63 for (char ch : name)
64 {
65 /* Convert character to lower case.
66 * lower
67 */
68 ch = tolower(ch);
69 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
70 }
71
72 return hashVal;

Line: 67, Col: 9

Name Value
_for_begin @0x7fffb2fc
_for_end @0x7fffb2ff
_for_range "AdaLovelac
ch 'a'
first "Ada"
hashVal 0
kLargePrime 15485863
kSmallPrime 137
last "Lovelace"

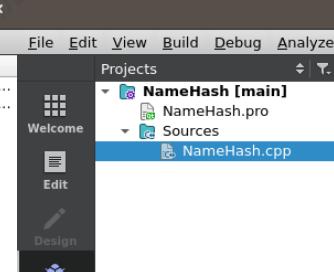
So what did that line of code do?

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555ab2e1	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::_Function_han			0x5555556037fc								
4	GThreadStd::run()			0x5555555e6616								
5	??			0x7ffffe4dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl) Issues Search Results Application Output Compile Output QML Debugger Console General Messages Version Control Test Results





did that line of code do?

NameHash.cpp @ NameHash [main] - Qt Creator

Tools Window Help

File Line: 67, Col: 9

```
50  /* This hashing scheme needs two prime numbers, a large prime
51   * prime. These numbers were chosen because their product is
52   * 2^31 - kLargePrime - 1.
53   */
54   static const int kLargePrime = 15485863;
55   static const int kSmallPrime = 137;
56
57   int hashVal = 0;
58
59   /* Iterate a
60    * name, upda
61    */
62   for (char ch
63     /* Conve
64     * lower
65     */
66     ch = tolower(ch);
67     hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68   }
69
70   return hashVal;
71 }
72 }
```

/* This hashing scheme needs two prime numbers, a large prime
* prime. These numbers were chosen because their product is
* $2^{31} - kLargePrime - 1$.
*/
static const int kLargePrime = 15485863;
static const int kSmallPrime = 137;

int hashVal = 0;

/* Iterate a
* name, upda
*/
for (char ch
 /* Conve
 * lower
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;
}

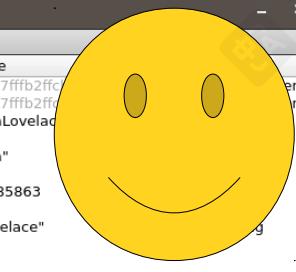
那行代码做了什么?

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555ab2e1	● 1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::_Function_han...			0x55555556037fc								
4	GThreadStd::run()			0x5555555e6616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results





代码做了

Qt Creator interface showing two side-by-side code editors for the file `NameHash.cpp`.

The code implements a hashing scheme:

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;
static const int kSmallPrime = 137;

int hashVal = 0;
/* Iterate a
 * name, upd
 */
for (char ch
    /* Conve
     * lower
    */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;
```

Annotations and screenshots:

- Left Screenshot:** A callout bubble highlights the line `ch = tolower(ch);`. The text inside the bubble reads:

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.
- Right Screenshot:** A callout bubble highlights the same line `ch = tolower(ch);`. The text inside the bubble reads:

这一行将 `ch` 转换为小写。The `tolower` 函数接收一个字符并返回其小写版本，所以这个会覆盖 `ch` 为其小写版本。将其自身的小写版本。

Both screenshots show the debugger at the end of the range step. The variable table shows the following values:

Name	Value	Type
for_begin	@0x7ffffb2ff0	
for_end	@0x7ffffb2ff0	
for_range	"AdaLovelac	
ch	'a'	
first	"Ada"	
hashVal	0	
kLargePrime	15485863	
kSmallPrime	137	
last	"Lovelace"	

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

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

您可以通过查看值来实际看到这一点侧面的面板！

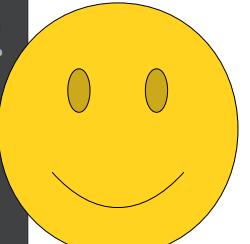
```
/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */ for (char ch: first + last) {    first: "Ada"    last: "Lovelace"    /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127. */    ch = tolower(ch);    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; }
```

```
return hashVal;
```

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */ for (char ch: first + last) { first: "Ada" last: "Lovelace" /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127. */ ch = tolower(ch); hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; }

```
return hashVal;
```

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



```
/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) {    first: "Ada"    last: "Lovelace"
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;
```

File Edit View Build Debug Analyze Tools Window Help

NameHash.cpp @ NameHash [main] - Qt Creator

Projects NameHash [main]

Source NameHash.cpp Sources NameHash.h

Line: 67, Col: 9

Name Value Type

for_begin @0x7ffb2ffcb78 std::string::iterator

for_end @0x7ffb2ffcb80 std::string::iterator

for_range "AdaLovelace" std::string &

ch 'a' char

first "Ada" std::string

hashVal 0 int

kLargePrime 15485863 int

kSmallPrime 137 int

last "Lovelace" std::string

0x61

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con/Ignore Threads

1 nameHash NameHash.cpp 67 0x5555555ab2e1 1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

2 studentMain NameHash.cpp 31 0x5555555ab0fb

3 std::__Function_han... 0x55555556037fc

4 GThreadStd::run() 0x55555555e6616

5 ?? 0x7ffff64dc2b3

6 start_thread pthread_create.c 442 0x7ffff6094b43

7 clone3 clone3.S 81 0x7ffff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

请注意，与 ch 关联的值已更改
从 'A' 到 'a' – 现在已变为小写！

```
/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) {    first: "Ada"    last: "Love...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;
```

Registers

Name	Value	Type
_for_begin	@0x7ffff2ffcb78	std::string::iterator
_for_end	@0x7ffff2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

Call Stack

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con Ignore	Threads
1	nameHash	NameHash.cpp	67	0x5555555ab2e1	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)	
2	studentMain	NameHash.cpp	31	0x5555555ab0fb							
3	std::Function::ha...			0x55555556037fc							
4	GThreadStd::run()			0x5555555e6616							
5	??			0x7ffff64dc2b3							
6	start_thread	pthread_create.c	442	0x7ffff6094b43							
7	clone3	clone3.S	81	0x7ffff6126a00							

NameHash.cpp @ NameHash [main] - Qt Creator

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

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */
for (char ch: first + last) { first: "Ada" last: "Lovelace"
 /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127.
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;

Line: 67, Col: 9

Name	Value	Type
__for_begin	@0x7fffb2ffcb78	std::string::iterator
for_end	@0x7fffb2ffcb80	std::string::iterator
for_range	"AdaLovelace"	std::string &
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

0x61

NameHash.cpp @ NameHash [main] - Qt Creator

如果你注意到, 这个值是红色的, 而所有其他的值都是黑色的。

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */
for (char ch: first + last) { first: "Ada" last: "Lovelace"
 /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127.
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;

Line: 67, Col: 9

Name	Value	Type
__for_begin	@0x7fffb2ffcb78	std::string::iterator
for_end	@0x7fffb2ffcb80	std::string::iterator
for_range	"AdaLovelace"	std::string &
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

0x61

NameHash.cpp @ NameHash [main] - Qt Creator

Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x5555555ab2e1	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::Function_h...			0x5555556037fc								
4	GThreadStd::run()			0x5555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x5555555ab2e1	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::Function_h...			0x5555556037fc								
4	GThreadStd::run()			0x5555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

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.

这表示这里的值自上一步以来已经改变。这是一种非常实用的方法，可程序。

```
/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */ for (char ch: first + last) {    first: "Ada"    last: "Lovelace"    /* Convert the input character to lower case. The numeric lower-case letters are always less than 127. */    ch = tolower(ch);    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; }
```

```
/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */ for (char ch: first + last) {    first: "Ada"    last: "Lovelace"    /* Convert the input character to lower case. The numeric lower-case letters are always less than 127. */    ch = tolower(ch);    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; }
```

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects Projects Projects

Welcome Welcome Welcome

Edit Edit Edit

Design Design Design

Debug Debug Debug

Projects Projects Projects

Welcome Welcome Welcome

Edit Edit Edit

Design Design Design

Debug Debug Debug

Projects Projects Projects

Welcome Welcome Welcome

Edit Edit Edit

Design Design Design

Debug Debug Debug

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

现在, 让我们看看第67行, 我们在那里

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */

for (char ch: first + last) { first: "Ada" last: "Love..." /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127. */ ch = tolower(ch); hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; }

return hashVal;

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */

for (char ch: first + last) { first: "Ada" last: "Love..." /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127. */ ch = tolower(ch); hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; }

return hashVal;

Line: 67, Col: 9 Line: 67, Col: 9 Line: 67, Col: 9

Name Value Type Name Value Type Name Value Type

Name	Value	Type
_for_begin	@0x7fffb2ffcb78	std::string::iterator
_for_end	@0x7fffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

Name	Value	Type
_for_begin	@0x7fffb2ffcb78	std::string::iterator
_for_end	@0x7fffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

Name	Value	Type
_for_begin	@0x7fffb2ffcb78	std::string::iterator
_for_end	@0x7fffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range". Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range". Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con Ignore Threads

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x5555555ab2e1	1	nameHash	NameHash.cpp	67	0x5555555ab2e1			(all)
2	studentMain	NameHash.cpp	31	0x5555555ab0fb	2	studentMain	NameHash.cpp	31	0x5555555ab0fb			
3	std::Function_h...			0x55555556037fc	3	std::Function_h...			0x55555556037fc			
4	GThreadStd::run()			0x5555555e616	4	GThreadStd::run()			0x5555555e616			
5	??			0x7ffff64dc2b3	5	??			0x7ffff64dc2b3			
6	start_thread	pthread_create.c	442	0x7ffff6094b43	6	start_thread	pthread_create.c	442	0x7ffff6094b43			
7	clone3	clone3.S	81	0x7ffff6126a00	7	clone3	clone3.S	81	0x7ffff6126a00			

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects Projects Projects

Welcome Welcome Welcome

Edit Edit Edit

Design Design Design

Debug Debug Debug

Projects Projects Projects

Help Help Help

Projects Projects Projects

Help Help Help

Projects Projects Projects

Help Help Help

Line: 67, Col: 9

Line: 67, Col: 9

Line: 67, Col: 9

Name Value Type

for_begin @0x7fffb2ffcb78 std::string::iterator

for_end @0x7fffb2ffcb80 std::string::iterator

for_range "AdaLovelace" std::string &

ch 'a' 97 char

first "Ada" std::string

hashVal 0 int

kLargePrime 15485863 int

kSmallPrime 137 int

last "Lovelace" std::string

0x61

Name Value Type

for_begin @0x7fffb2ffcb78 std::string::iterator

for_end @0x7fffb2ffcb80 std::string::iterator

for_range "AdaLovelace" std::string &

ch 'a' 97 char

first "Ada" std::string

hashVal 0 int

kLargePrime 15485863 int

kSmallPrime 137 int

last "Lovelace" std::string

0x61

Name Value Type

for_begin @0x7fffb2ffcb78 std::string::iterator

for_end @0x7fffb2ffcb80 std::string::iterator

for_range "AdaLovelace" std::string &

ch 'a' 97 char

first "Ada" std::string

hashVal 0 int

kLargePrime 15485863 int

kSmallPrime 137 int

last "Lovelace" std::string

0x61

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.

说实话，这行代码相当密集。它执行一些奇怪的数学计算，针对不同的值。

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */

for (char ch: first + last) { first: "Ada" last: "Lovelace" /* Convert the input character to lower case. The numeric lower-case letters are always less than 127. */ ch = tolower(ch), hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; }

return hashVal;

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */

for (char ch: first + last) { first: "Ada" last: "Lovelace" /* Convert the input character to lower case. The numeric lower-case letters are always less than 127. */ ch = tolower(ch), hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; }

return hashVal;

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 67 0x5555555ab2e1 1 nameHash(std::string, std::string)

2 studentMain NameHash.cpp 31 0x5555555ab0fb ...eHash.cpp 66 ...ab2d3 (all)

3 std::Function_h... 0x55555556037fc

4 GThreadStd::run() 0x5555555e616 0x7ffff64dc2b3

5 ?? 0x7ffff64dc2b3

6 start_thread pthread_create.c 442 0x7ffff6094b43 0x7ffff64dc2b3

7 clone3 clone3.S 81 0x7ffff6126a00 0x7ffff6094b43

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

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

从根本上说，但它只是计算一些值的奇怪函数，并将其存储到 hashVal.

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */
for (char ch: first + last) { first: "Ada" last: "Lovelace"
 /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127.
 */
 ch = tolower(ch),
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;

/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */
for (char ch: first + last) { first: "Ada" last: "Lovelace"
 /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127.
 */
 ch = tolower(ch),
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".
Level Function File Line Address Number Function File Line Address Con Ignore Threads
1 nameHash NameHash.cpp 67 0x5555555ab2e1 1 nameHash(std::string, std::string)
2 studentMain NameHash.cpp 31 0x5555555ab0fb ...eHash.cpp 66 ...ab2d3 (all)
3 std::Function_h... 0x5555556037fc
4 GThreadStd::run() 0x5555555e616 0x7fff64dc2b3
5 ?? 0x7fff64dc2b3
6 start_thread pthread_create.c 442 0x7fff6094b43 0x7fff64dc2b3
7 clone3 clone3.S 81 0x7ffff6126a00 0x7fff6094b43
Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".
Level Function File Line Address Number Function File Line Address Con Ignore Threads
1 nameHash NameHash.cpp 67 0x5555555ab2e1 1 nameHash(std::string, std::string)
2 studentMain NameHash.cpp 31 0x5555555ab0fb ...eHash.cpp 66 ...ab2d3 (all)
3 std::Function_h... 0x5555556037fc
4 GThreadStd::run() 0x5555555e616 0x7fff64dc2b3
5 ?? 0x7fff64dc2b3
6 start_thread pthread_create.c 442 0x7fff6094b43 0x7fff64dc2b3
7 clone3 clone3.S 81 0x7ffff6126a00 0x7fff6094b43

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects Projects Projects

Welcome Welcome Welcome

Edit Edit Edit

Design Design Design

Debug Debug Debug

Projects Projects Projects

Welcome Welcome Welcome

Edit Edit Edit

Design Design Design

Debug Debug Debug

Projects Projects Projects

Welcome Welcome Welcome

Edit Edit Edit

Design Design Design

Debug Debug Debug

?

?

?

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

让我们运行那行代码看看会发生什么

发生!

0 0

0 0

0 0

```
/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */  
for (char ch: first + last) {    first: "Ada"    last: "Love...  
    /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127.  
     */  
    ch = tolower(ch),  
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
  
    return hashVal;
```

```
/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */  
for (char ch: first + last) {    first: "Ada"    last: "Love...  
    /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127.  
     */  
    ch = tolower(ch),  
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
  
    return hashVal;
```

```
/* Iterate across all the characters in the first name, then * name, updating the hash at each step. */  
for (char ch: first + last) {    first: "Ada"    last: "Love...  
    /* Convert the input character to lower case. The numeric * lower-case letters are always less than 127.  
     */  
    ch = tolower(ch),  
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
  
    return hashVal;
```

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x5555555ab2e1	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::function<_H>			0x55555556037fc								
4	GThreadStd::run()			0x5555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;
static const int kSmallPrime = 137;

int hashVal = 0;

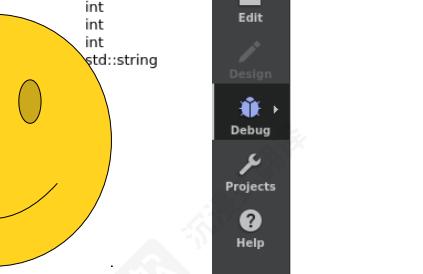
/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch : first) {
    hashVal = (hashVal * kSmallPrime + ch) % kLargePrime;
}
for (char ch : name) {
    hashVal = (hashVal * kSmallPrime + ch) % kLargePrime;
}

return hashVal;
}
```

Name Value Type

Name	Value	Type
_for_begin	@0x7fffb2ffcb78	std::string::iterator
_for_end	@0x7fffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'a' 97	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Love"	std::string

0x61



File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;
static const int kSmallPrime = 137;

int hashVal = 0;

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch : first) {
    hashVal = (hashVal * kSmallPrime + ch) % kLargePrime;
}
for (char ch : name) {
    hashVal = (hashVal * kSmallPrime + ch) % kLargePrime;
}

return hashVal;
}
```

Name Value Type

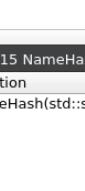
Name	Value	Type
_for_begin	@0x7fffb2ffcb78	std::string::iterator
_for_end	@0x7fffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'a' 97	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Love"	std::string

0x61



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

将鼠标悬停在“Step Over”按钮上，确认你所点击的按钮确实是“Step Over”，然后点击它！当你这样做时...



NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

```
File Edit View Build Debug Analyze Tools Window Help
```

```
File Edit View Build Debug Analyze Tools Window Help
```

Projects NameHash.cpp Line: 62, Col: 5

Projects NameHash.cpp Line: 62, Col: 5

```
50 /* This hashing scheme needs two prime numbers, a large prime  
51 * prime. These numbers were chosen because their product is  
52 *  $2^{31} - kLargePrime - 1$ .  
53 */  
54 static const int kLargePrime = 15485863; kLargePrime: 1...  
55 static const int kSmallPrime = 137; kSmallPrime: 137  
56  
57 int hashVal = 0; hashVal: 97  
58  
59 /* Iterate across all the characters in the first name, then  
60 * name, updating the hash at each step.  
61 */  
62 for (char ch: first + last) { ch: 97 first: "Ada" las...  
63 /* Convert the input character to lower case. The numeric  
64 * lower-case letters are always less than 127.  
65 */  
66 ch = tolower(ch);  
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime  
68 }  
69  
70 return hashVal;  
71  
72 }
```

Projects NameHash.cpp Line: 62, Col: 5

Projects NameHash.cpp Line: 62, Col: 5

```
50 /* This hashing scheme needs two prime numbers, a large prime  
51 * prime. These numbers were chosen because their product is  
52 *  $2^{31} - kLargePrime - 1$ .  
53 */  
54 static const int kLargePrime = 15485863; kLargePrime: 1...  
55 static const int kSmallPrime = 137; kSmallPrime: 137  
56  
57 int hashVal = 0; hashVal: 97  
58  
59 /* Iterate across all the characters in the first name, then  
60 * name, updating the hash at each step.  
61 */  
62 for (char ch: first + last) { ch: 97 first: "Ada" las...  
63 /* Convert the input character to lower case. The numeric  
64 * lower-case letters are always less than 127.  
65 */  
66 ch = tolower(ch);  
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime  
68 }  
69  
70 return hashVal;  
71  
72 }
```

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

...你最终会得到一些东西

就像这样!

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x5555555ab31b	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::Function_h...			0x5555556037fc								
4	GThreadStd::run()			0x5555555e616								
5	???			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 97

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 97 first: "Ada" las...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime
}

return has
```

Name Value Type

- _for_begin @0x7fff2ffcb78 std::string::iter...
- _for_end @0x7fff2ffcb80 std::string::iter...
- _for_range "AdaLovelace" std::string &
- ch 'a' 97 0x61 char std::string
- first "Ada" std::string
- hashVal 97 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 97

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 97 first: "Ada" las...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime
}

return has
```

Name Value Type

- _for_begin @0x7fff2ffcb78 std::string::iter...
- _for_end @0x7fff2ffcb80 std::string::iter...
- _for_range "AdaLovelace" std::string &
- ch 'a' 97 0x61 char std::string
- first "Ada" std::string
- hashVal 97 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

Let's see what's changed.

让我们看看有什么变化。

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x5555555ab31b	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::Function_h...			0x5555556037fc								
4	GThreadStd::run()			0x5555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7fff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

Name Hash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 97

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 97 first: "Ada" las...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime
}

return has
```

Name Value Type

- _for_begin @0x7fff2ffcb78 std::string::iter...
- _for_end @0x7fff2ffcb80 std::string::iter...
- _for_range "AdaLovelace" std::string &
- ch 'a' 97 0x61 char std::string
- first "Ada" std::string
- hashVal 97 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 97

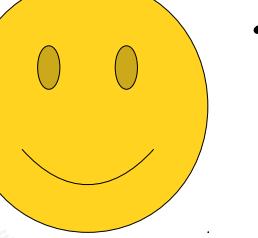
/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 97 first: "Ada" las...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime
}

return has
```

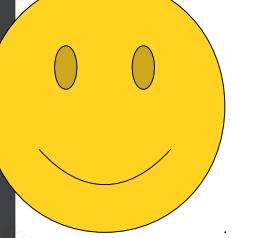
Name Value Type

- _for_begin @0x7fff2ffcb78 std::string::iter...
- _for_end @0x7fff2ffcb80 std::string::iter...
- _for_range "AdaLovelace" std::string &
- ch 'a' 97 0x61 char std::string
- first "Ada" std::string
- hashVal 97 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

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!



请注意存储在 `hashVal` 中的值
们知道它改变了，因为
为红色，并且我们知道其他任何东西都没有变化，因
为别的地方没有红色的东西！



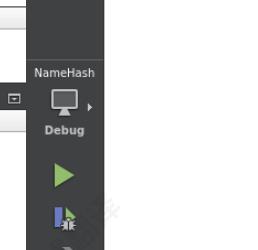
Second, notice that we're back up at the top of the for loop, since that's where the yellow arrow is pointing. We ended up back here because this is the next line that gets executed.

```
File Edit View Build Debug Analyze Tools Window Help
Projects NameHash [main] NameHash.cpp nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5
Name Hash Value Type
std::string::iterator std::string::iterator std::string::iterator
std::string::iterator std::string::iterator std::string::iterator
Sources NameHash.cpp
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137
int hashVal = 0; hashVal: 97
/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char
    /* Con
     * low
     */
    ch = t
    hashVa
}
return hashVal,
```

For loop variables and values:

Name	Value	Type
_for_begin	@0x7ffb2ffcb78	std::string::iterator
_for_end	@0x7ffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string::iterator
ch	'a'	char
first	"Ada"	std::string
hashVal	97	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

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.



其次, 请注意我们回到了
for 循环的顶部, 因为那里是黄色箭头
指向的地方。我们最终回到了这里, 因为这是
下一行要执行的代码。

```
NameHash.cpp @ NameHash [main] - Qt Creator
File Edit View Build Debug Analyze Tools Window Help
Projects NameHash [main]
NameHash.pro Sources NameHash.cpp Line: 62, Col: 5
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;    kLargePrime: 1...
static const int kSmallPrime = 137;          kSmallPrime: 137

int hashVal = 0;                          hashVal: 97

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char
     /* Con
      * low
      */
     ch = t
     hashVa
}
return hashVal;

NameValueType
__for_begin@0x7ffffb2ffcb78std::string::iter.
__for_end@0x7ffffb2ffcb80std::string::iter.
__for_range"AdaLovelace"std::string &
ch'a'970x61char
first"Ada"std::string
hashVal97int
kLargePrime15485863int
kSmallPrime137int
last"Lovelace"std::string
```

请注意我们回到了
循环的顶部，因为那里是黄色箭头
的地方。我们最终回到了这里，因
为这是下一行要执行的代码。



NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5

50 */ This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137
56
57 int hashVal = 0; hashVal: 97
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) { ch: 97 first: "Ada" las...
63 /* Convert the input character to lower case. The numeric
64 * lower-case letters are always less than 127.
65 */
66 ch = t
67 hashVa }
68
69 return hashva
70
71
72 }

50 */ This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137
56
57 int hashVal = 0; hashVal: 97
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) { ch: 97 first: "Ada" las...
63 /* Convert the input character to lower case. The numeric
64 * lower-case letters are always less than 127.
65 */
66 ch = t
67 hashVa }
68
69 return hashva
70
71
72 }

50 */ This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137
56
57 int hashVal = 0; hashVal: 97
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) { ch: 97 first: "Ada" las...
63 /* Convert the input character to lower case. The numeric
64 * lower-case letters are always less than 127.
65 */
66 ch = t
67 hashVa }
68
69 return hashva
70
71
72 }

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

我们刚刚单步执行了单次迭代
那个循环！真酷！

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address

1 nameHash NameHash.cpp 62 0x5555555ab31b 1 nameHash(std::string, std::string)

2 studentMain NameHash.cpp 31 0x5555555ab0fb 2 studentMain

3 std::Function_h... 0x5555556037fc 3 std::Function_h...

4 GThreadStd::run() 0x5555555e616 4 GThreadStd::run()

5 ?? 0x7ffff64dc2b3 5 ??

6 start_thread pthread_create.c 442 0x7ffff6094b43 6 start_thread

clone3 pthread_create.c 81 0x7ffff6126a00 7 clone3.S

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5

Welcome Edit Design Debug Projects Help

Welcome Edit Design Debug Projects Help

Welcome Edit Design Debug Projects Help

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 97

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 97 first: "Ada" las...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = t
    hashVa
}

return hashva
```

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

Name Value Type
__for_begin @0x7fffb2ffcb78 std::string::iter...
__for_end @0x7fffb2ffcb80 std::string::iter...
__for_range "AdaLovelace" std::string &
ch 'a' 97 char
first "Ada" std::string
hashVal 97 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

0x61 char std::string int std::string int std::string

0x61 char std::string & std::string & int int std::string

0x61 char std::string & std::string & int int std::string

int hashVal = 0; hashVal: 97

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 97 first: "Ada" las...
 /* Convert the input character to lower case. The numeric
 * lower-case letters are always less than 127.
 */
 ch = t
 hashVa
}

return hashva

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

Let's go do it again!

让我们再试一次!

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address

1 nameHash NameHash.cpp 62 0x5555555ab31b 1 nameHash(std::string, std::string)

2 studentMain NameHash.cpp 31 0x5555555ab0fb 2 studentMain

3 std::Function_h... 0x55555556037fc 3 std::Function_h...

4 GThreadStd::run() 0x5555555e616 4 GThreadStd::run()

5 ?? 0x7ffff64dc2b3 5 ??

6 start_thread pthread_create.c 442 0x7ffff6094b43 6 start_thread

7 clone3 clone3.S 81 0x7ffff6126a00 7 clone3

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp Line: 62, Col: 5

Projects NameHash.cpp Line: 62, Col: 5

Projects NameHash.cpp Line: 62, Col: 5

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 97

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 97 first: "Ada" las...
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = t
    hashVa
}

return hashva
```

/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: 97

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 97 first: "Ada" las...
 /* Convert the input character to lower case. The numeric
 * lower-case letters are always less than 127.
 */
 ch = t
 hashVa

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

再次, 将鼠标移到 Step Over 按钮上 (并确保它显示 "Step Over" 和而不是其他东西!), 然后点击它。

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

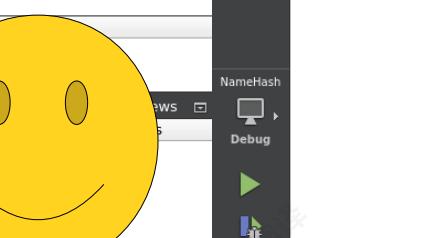
Level Function File Line Address Number Function File Line Address

1	nameHash	NameHash.cpp	62	5555ab31b	1	nameHash(std::string, std::string)	NameHash.cpp	66	...
2	studentMain	NameHash.cpp	31	5555ab0fb					
3	std::Function_h...			56037fc					
4	GThreadStd::run()			56037fc					
5	??			56037fc					
6	start_thread	pthread_create.c	442	0x7ffff6094b43					
7	clone3	clone3.S	81	0x7ffff6126a00					

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results



NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp Line: 66, Col: 9

Projects NameHash.cpp Line: 66, Col: 9

Projects NameHash.cpp Line: 66, Col: 9

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137
```

int hashVal

/* Iterate
 * name, up
 */
for (char ch: first + last) { first: "Ada" last: "Lovelace"
 /* Convert the input character to lower case. The numeric
 * lower-case letters are always less than 127.
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

Name Value Type

Name Value Type

Name Value Type

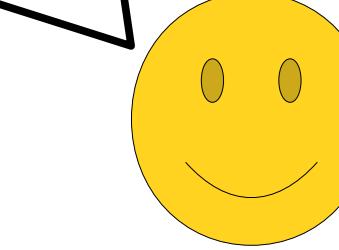
Name	Value	Type
for_begin	@0x7fff2ffcb78	std::string::iterator
for_end	@0x7fff2ffcb80	std::string::iterator
for_range	Ada Lovelace	std::string &
ch	'd'	char
first	"Ad"	std::string
hashVal	97	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

Name	Value	Type
for_begin	@0x7fff2ffcb78	std::string::iterator
for_end	@0x7fff2ffcb80	std::string::iterator
for_range	Ada Lovelace	std::string &
ch	'd'	char
first	"Ad"	std::string
hashVal	97	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

Name	Value	Type
for_begin	@0x7fff2ffcb78	std::string::iterator
for_end	@0x7fff2ffcb80	std::string::iterator
for_range	Ada Lovelace	std::string &
ch	'd'	char
first	"Ad"	std::string
hashVal	97	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

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

现在我们到了! 注意 ch 现在的值 'd', 这是名字 Ada 的第二个字母。



NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp Line: 66, Col: 9

Projects NameHash.cpp Line: 66, Col: 9

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137
```

Name Value Type

_for_begin	@0x7fffb2ffcb78	std::string::iterator
_for_end	@0x7fffb2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'd'	char
first	"Ada"	std::string
hashVal	97	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

int hashVal = 1;
/* Iterate over each character in the name, updating the hash value.
 * name, up to last.
 */
for (char ch: first + last) { first: "Ada" last: "Lovelace"
 /* Convert the input character to lower case. The numeric
 * lower-case letters are always less than 127.
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;

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

int hashVal = 1;
/* Iterate over each character in the name, updating the hash value.
 * name, up to last.
 */
for (char ch: first + last) { first: "Ada" last: "Lovelace"
 /* Convert the input character to lower case. The numeric
 * lower-case letters are always less than 127.
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;

点点击“Step Over”再次运行这一行 code.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped at breakpoint 1 in thread 15.

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1	nameHash	NameHash.cpp	66	0x5555ab2d3	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)
2	studentMain	NameHash.cpp	31	0x5555ab0fb	2	studentMain	NameHash.cpp	31	0x5555ab0fb	
3	std::Function_h...			0x56037fc	3	std::Function_h...			0x56037fc	
4	GThreadStd::run()			0x7fff6126a16	4	GThreadStd::run()			0x7fff6126a16	
5	??			0x7fff6126a00	5	??			0x7fff6126a00	
6	start_thread	pthread_create.c	442	0x7ffff6094b43	6	start_thread	pthread_create.c	442	0x7ffff6094b43	
7	clone3	clone3.S	81	0x7ffff6126a00	7	clone3	clone3.S	81	0x7ffff6126a00	

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp Line: 67, Col: 9

Projects NameHash.cpp Line: 67, Col: 9

Projects NameHash.cpp Line: 67, Col: 9

```
/* This hashing scheme needs two prime numbers, a large prime.  
 * These numbers were chosen because their product is  
 * 2^31 - kLargePrime - 1.  
 */  
static const int kLargePrime = 15485863;  
  
static const int kSmallPrime = 137;  
  
int hashVal = 97;  
  
/* Iterate over the string from first to last.  
 * name, up to last inclusive.  
for (char ch: first + last) {    first: "Ada"    last: "Love...  
    /* Convert the input character to lower case. The numeric  
     * value of lower-case letters are always less than 127.  
     */  
    ch = tolower(ch);  
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;  
  
}  
  
return hashVal;
```

/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;

static const int kSmallPrime = 137;

int hashVal = 97;

/* Iterate over the string from first to last.
 * name, up to last inclusive.
for (char ch: first + last) { first: "Ada" last: "Love...
 /* Convert the input character to lower case. The numeric
 * value of lower-case letters are always less than 127.
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

}

return hashVal;

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp Line: 67, Col: 9

Projects NameHash.cpp Line: 67, Col: 9

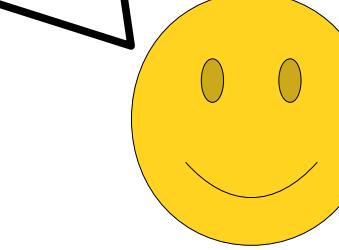
Projects NameHash.cpp Line: 67, Col: 9

Name	Value	Type
_for_begin	@0x7fff2ffcb78	std::string::iterator
_for_end	@0x7fff2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'd'	char
first	"Ada"	std::string
hashVal	97	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

Name	Value	Type
_for_begin	@0x7fff2ffcb78	std::string::iterator
_for_end	@0x7fff2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'd'	char
first	"Ada"	std::string
hashVal	97	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

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'.

你现在应该在这里。请注意，没有一个值发生了变化。这很合理，因为我们所做的一切是将一个 lower-case 'd' 转换为 lower-case 'd'。



NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Projects Help

Projects NameHash [main] Sources NameHash.cpp

50 /* This hashing scheme needs two prime numbers, a large prime 51 * prime. These numbers were chosen because their product is 52 * $2^{31} - kLargePrime - 1$. 53 */ 54 static const int kLargePrime = 15485863; 55 static const int kSmallPrime = 137; 56 57 int hashVal = 97; 58 59 /* Iterate over each character in the string. 60 * name, updating the hash value. 61 */ 62 for (char ch: first + last) { 63 /* Convert the input character to lower case. The numeric 64 * lower-case letters are always less than 127. 65 */ 66 ch = tolower(ch); 67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime; 68 } 69 70 return hashVal; 71 72 }

Line: 67, Col: 9

Name Value Type
for_begin @0x7ffffb2ffcb78 std::string::iterator
for_end @0x7ffffb2ffcb80 std::string::iterator
for_range "AdaLovelace" std::string
ch 'd' 100 char
first "Ada" std::string
hashVal 97 int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

0x64

Now, click "Step Over" one more time.

Step Over

Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 67 0x7ffffb2ffcb78 1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

2 studentMain NameHash.cpp 31 0x7ffffb2ffcb80 std::function<void()> 3 std::_Function_han... NameHash.cpp 31 0x7ffffb2ffcb80 std::function<void()> 4 GThreadStd::run() 5 ?? 6 start_thread pthread_create.c 442 0x7ffffb2ffcb80 std::function<void()> 7 clone3 pthread_create.c 442 0x7ffffb2ffcb80 std::function<void()> clone3.S 81 0x7ffffb2ffcb80 std::function<void()>

Type to locate (Ctrl) Issues Search Results Application Output Compile Output QML Debugger Console General Messages Version Control Test Results

Now, click "Step Over" one more time.

Step Over

Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 67 0x7ffffb2ffcb78 1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

2 studentMain NameHash.cpp 31 0x7ffffb2ffcb80 std::function<void()> 3 std::_Function_han... NameHash.cpp 31 0x7ffffb2ffcb80 std::function<void()> 4 GThreadStd::run() 5 ?? 6 start_thread pthread_create.c 442 0x7ffffb2ffcb80 std::function<void()> 7 clone3 pthread_create.c 442 0x7ffffb2ffcb80 std::function<void()> clone3.S 81 0x7ffffb2ffcb80 std::function<void()>

The screenshot shows the Qt Creator IDE interface with the following details:

- File Menu:** File, Edit, View, Build, Debug, Analyze, Tools, Window, Help.
- Projects View:** Shows the project "NameHash [main]" with files "NameHash.pro" and "Sources" containing "NameHash.cpp".
- Code Editor:** Displays the "NameHash.cpp" source code. The current line is 67, which contains the assignment `hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;`. A red box highlights the "Step Over" button in the toolbar below the editor.
- Variables View:** Shows the state of variables at line 67. The variable "hashVal" is highlighted in blue. Other variables shown include `_for_begin`, `_for_end`, `_for_range`, `ch`, `first`, `hashVal`, `kLargePrime`, `kSmallPrime`, and `last`.
- Callouts:** A large callout bubble points from a yellow smiley face icon to the "Step Over" button in the toolbar.
- Toolbar:** Includes buttons for Run, Stop, Step Into, Step Over (highlighted with a red box), and Step Out.
- Bottom Status Bar:** Shows tabs for Issues, Search Results, Application Output, Compile Output, QML Debugger Console, General Messages, Version Control, and Test Results.

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

```
File Edit View Build Debug Analyze Tools Window Help
```

```
File Edit View Build Debug Analyze Tools Window Help
```

Projects NameHash.cpp Line: 62, Col: 5

Projects NameHash.cpp Line: 62, Col: 5

```
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
```

```
/* This hashing scheme needs two prime numbers. These numbers were chosen because 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863;
static const int kSmallPrime = 137;

int hashVal = 0;
```

Look here!

Name Value Type

- _for_begin @0x7ffffb2ffcb78 std::string::iterator
- _for_end @0x7ffffb2ffcb80 std::string::iterator
- _for_range "AdaLovelace" std::string &
- ch 'd' std::string
- first "Ada" int
- hashVal 15485863 int
- kLargePrime 15485863 int
- kSmallPrime 137 int
- last "Lovelace" std::string

hashVal: ?????

hashVal: ?????

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!**

看这里!

您现在将处于程序的这一点。我们已经在这个图像中遮盖了hashVal 的值，因为在这个阶段，您应该能够通过读取侧边栏中的值。**这是特殊值**

我们希望您在提交作业时告诉我们！

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x5555555ab31b	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab
2	studentMain	NameHash.cpp	31	0x5555555ab0fb					
3	std::Function<_Func_>::call			0x55555556037fc					
4	GThreadStd::run()			0x5555555e616					
5	???			0xfffff64dc2b3					
6	start_thread	pthread_create.c	442	0x7ffff6094b43					
7	clone3	clone3.S	81	0x7ffff6126a00					

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

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.

来完成这个调试器部分，我们想展示给你两个最后的小技巧，你可能会觉得有用在调试程序时。

```
/* This hashing scheme needs two prime numbers, a large prime. These numbers were chosen because their product is * 2^31 - kLargePrime - 1.
```

```
for (char ch: first + last) {    ch: 100    first: "Ada"    /* Convert the input character to lower case. The numeric lower-case letters are always less than 127. */    ch = tolower(ch);    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;}
```

```
return hashVal;
```

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555ab31b	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x555555ab0fb								
3	std::Function_h...			0x5555556037fc								
4	GThreadStd::run()			0x555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

/* This hashing scheme needs two prime numbers, a large prime. These numbers were chosen because their product is * 2^31 - kLargePrime - 1.

for (char ch: first + last) { ch: 100 first: "Ada" /* Convert the input character to lower case. The numeric lower-case letters are always less than 127. */ ch = tolower(ch); hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;}

return hashVal;

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555ab31b	1	nameHash(std::string, std::string)	...eHash.cpp	66	...ab2d3	(all)		
2	studentMain	NameHash.cpp	31	0x555555ab0fb								
3	std::Function_h...			0x5555556037fc								
4	GThreadStd::run()			0x555555e616								
5	??			0x7fff64dc2b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp Line: 62, Col: 5

Projects NameHash.cpp Line: 62, Col: 5

Projects NameHash.cpp Line: 62, Col: 5

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.
 */

for (char ch: first + last) { ch: 100 first: "Ada"
/* Convert the input character to lower case. The numeric
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;

Name Value Type

_for_begin @0x7fffb2ffcb78 std::string::iterator
_for_end @0x7fffb2ffcb80 std::string::iterator
_for_range "AdaLovelace" std::string &
ch 'd' 100 char
first "Ada" std::string int
hashVal ??? std::string int
kLargePrime 15485863 int

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * $2^{31} - kLargePrime - 1$.
 */

for (char ch: first + last) { ch: 100 first: "Ada"
/* Convert the input character to lower case. The numeric
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

return hashVal;

Name Value Type

_for_begin @0x7fffb2ffcb78 std::string::iterator
_for_end @0x7fffb2ffcb80 std::string::iterator
_for_range "AdaLovelace" std::string &
ch 'd' 100 char
first "Ada" std::string int
hashVal ??? std::string int
kLargePrime 15485863 int

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

要开始这个, 请点击我们之前在程序中设置的断点
earlier in the program. If you do...

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 62 0x555555ab31b 1 nameHash(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

2 studentMain NameHash.cpp 31 0x555555ab0fb 2 studentMain(std::string, std::string) ...eHash.cpp 66 ...ab2d3 (all)

3 std::Function_h... 0x5555556037fc 3 std::Function_h...(std::function<void()>) ...eHash.cpp 66 ...ab2d3 (all)

4 GThreadStd::run() 0x555555e616 4 GThreadStd::run() ...eHash.cpp 66 ...ab2d3 (all)

5 ?? 5 ?? ...eHash.cpp 66 ...ab2d3 (all)

6 start_thread pthread_create.c 442 0x7ffff6094b43 6 start_thread(pthread_create) ...eHash.cpp 66 ...ab2d3 (all)

7 clone3 clone3.S 81 0x7ffff6126a00 7 clone3(clone3) ...eHash.cpp 66 ...ab2d3 (all)

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp Line: 62, Col: 5

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
for (char ch: first + last) {    ch: 100    first: "Ada"
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
```

... 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!

... 它应该清除断点。现在，如果我们再次以调试模式运行这个程序，它将不会在这个点停止，因为没有东西告诉它要停！

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1	nameHash	NameHash.cpp	62	0x555555ab31b				
2	studentMain	NameHash.cpp	31	0x555555ab0fb				
3	std::Function<_ha...			0x55555556037fc				
4	GThreadStd::run()			0x5555555e616				
5	???			0xfffff64dc2b3				
6	start_thread	pthread_create.c	442	0x7ffff6094b43				
7	clone3	clone3.S	81	0xfffff6126a00				

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp Line: 62, Col: 5

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
for (char ch: first + last) {    ch: 100    first: "Ada"
    /* Convert the input character to lower case. The numeric
     * lower-case letters are always less than 127.
     */
    ch = tolower(ch);
    hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
```

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con Ignore Threads

1	nameHash	NameHash.cpp	62	0x555555ab31b				
2	studentMain	NameHash.cpp	31	0x555555ab0fb				
3	std::Function<_ha...			0x55555556037fc				
4	GThreadStd::run()			0x5555555e616				
5	???			0xfffff64dc2b3				
6	start_thread	pthread_create.c	442	0x7ffff6094b43				
7	clone3	clone3.S	81	0xfffff6126a00				

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Projects NameHash [main] NameHash.pro Sources NameHash.cpp

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

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */

54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137

56
57 int hashVal = 0; hashVal: ?????

58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */

62 for (char ch: first + last) { ch: 100 first: "AdaLovelace"
63 /* Convert the input character to lower case.
64 * lower.
65 */
66 ch = t
67 hashVa
68 }
69
70 return has
71
72 }

/* This hashing scheme needs two prime numbers, a large prime
* prime. These numbers were chosen because their product is
* $2^{31} - kLargePrime - 1$.
*/

static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: ?????

/* Iterate across all the characters in the first name, then
name, updating the hash at each step.
*/

for (char ch: first + last) { ch: 100 first: "AdaLovelace"
/* Convert the input character to lower case.
* lower.
*/
ch = t
hashVa

Now, take a look back at these buttons.

NameValueType
_for_begin@0x7ffb2ffcb78std::string::iter
_for_end@0x7ffb2ffcb80std::string::iterator
_for_range"AdaLovelace"char
ch'd'100std::string
first"Ada"?????int
hashVal?????int
kLargePrime15485863int
kSmallPrime137int
last"Lovelace"std::string

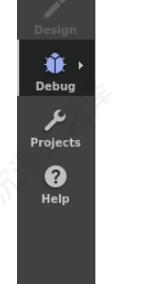
0x64

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range". Views

Level Function File Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 62 0x5555555ab31b
2 studentMain NameHash.cpp 31 0x5555555ab0fb
3 std::Function_h... 0x5555556037fc
4 GThreadStd::run() 0x5555555e6616
5 ?? 0x7ffff64dc2b3
6 start_thread pthread_create.c 442 0x7ffff6094b43
7 clone3 clone3.S 81 0x7ffff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results



, take a look back at these buttons.

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50 /* This hashing scheme needs two prime numbers, a large prime
51 * prime. These numbers were chosen because their product is
52 * $2^{31} - kLargePrime - 1$.
53 */
54 static const int kLargePrime = 15485863; kLargePrime: 1...
55 static const int kSmallPrime = 137; kSmallPrime: 137
56
57 int hashVal = 0; hashVal: ?????
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) { ch: 100 first: "Ada"
63 /* Convert the input character to lower case.
64 * lower.
65 */
66 ch = t
67 hashVa
68 }
69
70 return has
71
72 }

现在，回头看看这些按钮。

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

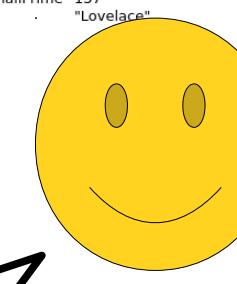
Level Function File Number Function File Line Address Con Ignore Threads

1 nameHash NameHash.cpp 62 0x5555555ab31b
2 studentMain NameHash.cpp 31 0x5555555ab0fb
3 std::Function_h...
4 GThreadStd::run()
5 ??
6 start_thread pthread_create.c 442 0x7ffff6094b43
7 clone3 clone3.S 81 0x7ffff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Name Value Type
__for_begin @0x7fff2ffcb78 std::string::iterator
__for_end @0x7fff2ffcb80 std::string::iterator
__for_range "AdaLovelace" std::string &
ch 'd' 100 char
first "Ada" std::string
hashVal ????? int
kLargePrime 15485863 int
kSmallPrime 137 int
last "Lovelace" std::string

0x64



回头看看这些

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: ??????

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 100 first: "Ada"
    /* Convert the input character to lower case
     * lower
     */
    ch = t
    hashVa
}

return has
```

Name Value Type

Name	Value	Type
_for_begin	@0x7fff2ffcb78	std::string::iterator
_for_end	@0x7fff2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'd'	char
first	"Ada"	std::string
hashVal	?????	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime.
 * These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: ??????

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 100 first: "Ada"
    /* Convert the input character to lower case
     * lower
     */
    ch = t
    hashVa
}

return has
```

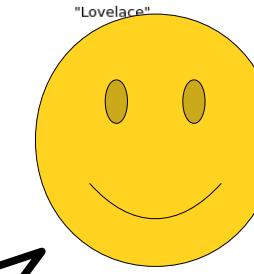
Name Value Type

Name	Value	Type
_for_begin	@0x7fff2ffcb78	std::string::iterator
_for_end	@0x7fff2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'd'	char
first	"Ada"	std::string
hashVal	?????	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

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

将鼠标悬停在最右边的那个上面。当你将鼠标悬停在上面时，它应该显示“退出步骤。”



NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: ??????

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 100 first: "Ada"
    /* Convert the input character to lower case
     * lower
     */
    ch = t
    hashVa
}

return has
```

Name Value Type

Name	Value	Type
_for_begin	@0x7fff2ffcb78	std::string::iterator
_for_end	@0x7fff2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'd'	char
first	"Ada"	std::string
hashVal	?????	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: ??????

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 100 first: "Ada"
    /* Convert the input character to lower case
     * lower
     */
    ch = t
    hashVa
}

return has
```

Name Value Type

Name	Value	Type
_for_begin	@0x7fff2ffcb78	std::string::iterator
_for_end	@0x7fff2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'd'	char
first	"Ada"	std::string
hashVal	?????	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] Sources NameHash.cpp

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: ??????

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 100 first: "Ada"
    /* Convert the input character to lower case
     * lower
     */
    ch = t
    hashVa
}

return has
```

Name Value Type

Name	Value	Type
_for_begin	@0x7fff2ffcb78	std::string::iterator
_for_end	@0x7fff2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'd'	char
first	"Ada"	std::string
hashVal	?????	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con Ignore Threads

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x5555555ab31b								
2	studentMain	NameHash.cpp	31	0x5555555ab0fb								
3	std::Function_h...			0x55555556037fc								
4	GThreadStd::run()			0x5555555e616								
5	??			0xfffff64dc2b3								
6	start_thread	pthread_create.c	442	0xfffff6094b43								
7	clone3	clone3.S	81	0xfffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 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.

不要立即点击。但当你点击时，
它将运行剩余的 `nameHash`
函数，直到它完成并返回。

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5

Projects NameHash.cpp # nameHash(string, string) -> int Unix (LF) Line: 62, Col: 5

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: ??????

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 100 first: "Ada"
    /* Convert the input character to lower case
     * lower
     */
    ch = t
    hashVa
}

return has
```

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

Name Value Type

Name	Value	Type
_for_begin	@0x7fff2ffcb78	std::string::iterator
_for_end	@0x7fff2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'd'	char
first	"Ada"	std::string
hashVal	?????	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

```
/* This hashing scheme needs two prime numbers, a large prime
 * prime. These numbers were chosen because their product is
 * 2^31 - kLargePrime - 1.
 */
static const int kLargePrime = 15485863; kLargePrime: 1...
static const int kSmallPrime = 137; kSmallPrime: 137

int hashVal = 0; hashVal: ??????

/* Iterate across all the characters in the first name, then
 * name, updating the hash at each step.
 */
for (char ch: first + last) { ch: 100 first: "Ada"
    /* Convert the input character to lower case
     * lower
     */
    ch = t
    hashVa
}

return has
```

现在，去点击那个按钮。如果你已经做对了所有的事情...

Name Value Type

Name	Value	Type
_for_begin	@0x7fff2ffcb78	std::string::iterator
_for_end	@0x7fff2ffcb80	std::string::iterator
_for_range	"AdaLovelace"	std::string &
ch	'd'	char
first	"Ada"	std::string
hashVal	?????	int
kLargePrime	15485863	int
kSmallPrime	137	int
last	"Lovelace"	std::string

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".

Level Function File Line Address Number Function File Line Address Con Ignore Threads

Level	Function	File	Line	Address	Number	Function	File	Line	Address	Con	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x5555ab31b								
2	studentMain	NameHash.cpp	31	0x5555ab0fb								
3	std::_Function_h...			0x5555ab0fb								
4	GThreadStd::run()			0x5555ab616								
5	??			0x7ffff6404b3								
6	start_thread	pthread_create.c	442	0x7ffff6094b43								
7	clone3	clone3.S	81	0x7ffff6126a00								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

```
File Edit View Build Debug Analyze Tools Window Help
```

```
File Edit View Build Debug Analyze Tools Window Help
```

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

20 using namespace std;

21

22 /* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.

23 */

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); hashValue: 0 fi

32

33 cout << "The hash of your name is: " << hashValue;

34 return 0;

35 }

36

37 /* This is the

38 * to talk more

39 * the meantime

40 * of the input

41 *

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 255.

255 int

Name Value Type

first "Ada" std::string

hashValue 0 int

last "Lovelace" std::string

Welcome Edit Design Debug Projects Help

File Edit View Build Debug Analyze Tools Window Help

```
File Edit View Build Debug Analyze Tools Window Help
```

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

20 using namespace std;

21

22 /* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.

23 */

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); hashValue: 0 fi

32

33 cout << "The hash of your name is: " << hashValue;

34 return 0;

35 }

36

37 /* This is the

38 * to talk more

39 * the meantime

40 * of the input

41 *

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 255.

255 int

Name Value Type

first "Ada" std::string

hashValue 0 int

last "Lovelace" std::string

Welcome Edit Design Debug Projects Help

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

...你应该得到一些东西 看起来像这样!

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "function-finished".

Level Function File Line Address Number Function

1 studentMain NameHash.cpp 31 0x5555555ab0fb

2 std::Function_h...

3 GThreadStd::run()

4 ??

5 start_thread pthread_create.c 442 0x7fff6094b43

6 clone3 clone3.S 81 0x7fff6126a00

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "function-finished".

Level Function File Line Address Number Function

1 studentMain NameHash.cpp 31 0x5555555ab0fb

2 std::Function_h...

3 GThreadStd::run()

4 ??

5 start_thread pthread_create.c 442 0x7fff6094b43

6 clone3 clone3.S 81 0x7fff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

```
using namespace std;  
  
/* Prototype for the nameHash function. This lets us use the function  
 * 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);    hashValue: 0  fi  
  
    cout << "The hash of your name is: " << hashValue;  
    return 0;  
}  
  
/* This is the  
 * to talk more  
 * the meantime  
 * of the input  
 *  
 * For those of you who are more mathematically inclined, this function  
 * treats each character in the input name as a number between 0 and 255.
```

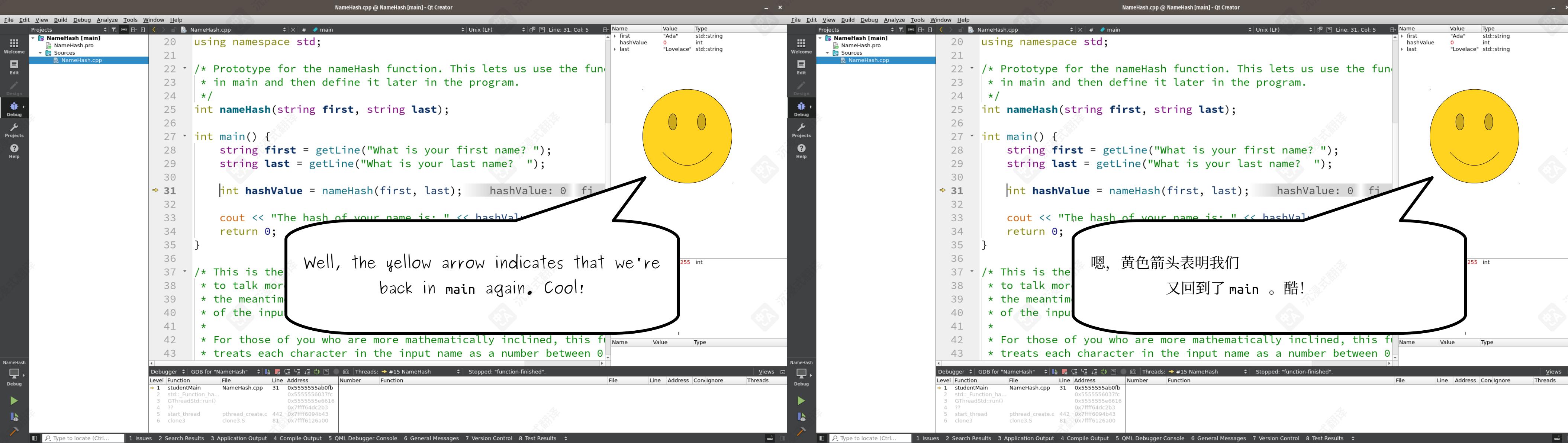
NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

```
using namespace std;  
  
/* Prototype for the nameHash function. This lets us use the function  
 * 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);    hashValue: 0  fi  
  
    cout << "The hash of your name is: " << hashValue;  
    return 0;  
}  
  
/* This is the  
 * to talk more  
 * the meantime  
 * of the input  
 *  
 * For those of you who are more mathematically inclined, this function  
 * treats each character in the input name as a number between 0 and 255.
```

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

让我们花一分钟来弄清楚状况。
我们究竟在哪里？



using namespace std;

/* Prototype for the nameHash function. This lets us use the function 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); // hashValue: 0 if you type "Ada Lovelace"

cout << "The hash of your name is: " << hashValue;

return 0;

}

/* This is the part where we talk more about the meaning of the input names. It's meant to be a fun exercise for those who are more mathematically inclined. It treats each character in the input name as a number between 0 and 255.

For those of you who are more mathematically inclined, this function

treats each character in the input name as a number between 0 and 255.

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

Name Value Type

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

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

using namespace std;

/* Prototype for the nameHash function. This lets us use the function 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); // hashValue: 0 if you type "Ada Lovelace"

cout << "The hash of your name is: " << hashValue;

return 0;

}

/* This is the part where we talk more about the meaning of the input names. It's meant to be a fun exercise for those who are more mathematically inclined. It treats each character in the input name as a number between 0 and 255.

For those of you who are more mathematically inclined, this function

treats each character in the input name as a number between 0 and 255.

嗯，黄色箭头表明我们又回到了 main。酷！

Name Value Type

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

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Qt Creator Window 1 (Left):

We can see that the `nameHash` function returned **15058255**. 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.)

Qt Creator Window 2 (Right):

我们可以看到`nameHash` 函数返回了**15058255**。谢谢，调试器！

(注意：在有些Mac上，这个数字可能无法显示。如果你看不到它——没关系，按正常流程继续即可。)

The code in both windows is as follows:

```
using namespace std;

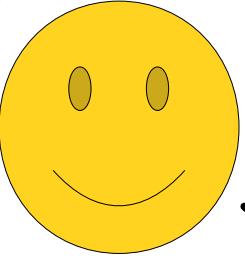
/* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.
 */
int nameHash(string first, string last);

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

    int hashValue = nameHash(first, last);    hashValue: 0  fi...
}

33   << "The hash of your name is: " << hashValue << endl;
```

A callout bubble from the left window points to the value `hashValue` in the code at line 33. A callout bubble from the right window points to the same value. Both bubbles point to the same line of code: `<< "The hash of your name is: " << hashValue << endl;`. A blue box highlights the value `15058255` in the tooltip.



But if we look up over here, we see that `hashValue` isn't storing 15058255, even though that's what was returned.

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

```
using namespace std;

/* Prototype for the nameHash function. This lets us use the function 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);      hashValue: 0 fi...
    cout << "The hash of your name is: " << hashValue << endl;
    0;
}
```

Name	Value	Type
first	Aus	std::string
hashValue	0	int
last	"placeholder"	std::string

returned value 15058255 int

Name	Value	Type
first	0	

File Line Address Configuration Views

The screenshot shows the Qt Creator IDE interface. On the left, there's a vertical toolbar with icons for Welcome, Edit, Design, Projects, and Help. The main area displays a C++ code editor for 'NameHash.cpp' with the following code:

```
using namespace std;  
  
/* Prototype for the nameHash function. This lets us use the function  
 * 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);    hashValue: 0 fi...  
  
    cout << "The hash of your name is: " << hashValue << endl;  
}  
  
A large yellow smiley face icon is positioned on the left side of the code editor. A black arrow points from the smiley face towards the variable declaration 'int hashValue = nameHash(first, last);'. A callout bubble with a black border contains the Chinese text: '但如果我们往这边看，我们会发现hashValue 并没有存储 15058255，尽管这就是返回的结果。 (你可能会在你的系统上看到0以外的数字——那没关系。)' (But if we look here, we will find that hashValue has not stored 15058255, even though this is the return result. (You might see other numbers than 0 on your system — that's fine.))
```

The screenshot shows the Qt Creator IDE interface. On the left, there's a vertical toolbar with icons for Welcome, Edit, Design, Projects, and Help. The main area has a title bar "NameHash.cpp @ NameHash [main] - Qt Creator". The menu bar includes File, Edit, View, Build, Debug, Analyze, Tools, Window, and Help. The Projects tab shows a project named "NameHash [main]" with files "NameHash.pro" and "Sources". The Sources tab displays the "NameHash.cpp" file content:

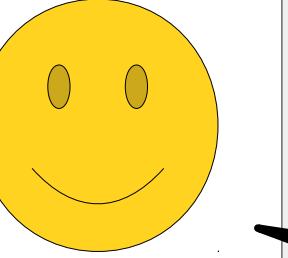
```
using namespace std;  
/* Prototype for the nameHash function. This lets us use the function  
 * in main and then define it later in the program.  
int nameHash(string first, string last);  
  
int main() {  
    string first = getLine("What is your first name? ");  
    string last = getLine("What is your last name? ");  
  
    int hashValue = nameHash(first, last); // Line 31  
  
    cout << "The hash of your name is: " << hashValue << endl;  
}
```

A large yellow smiley face icon is overlaid on the left side of the code editor. A callout bubble points from the line "int hashValue = nameHash(first, last);" to a handwritten note in the center of the screen.

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

We're quartering up the this function between 0 and 15058255.

The bottom of the screen shows the debugger stack trace and the variable inspector.



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

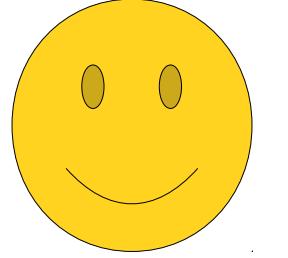
The screenshot shows the Qt Creator IDE interface. On the left, there's a vertical toolbar with icons for Welcome, Edit, Design, Projects, and Help. The main area has a title bar "NameHash.cpp @ NameHash [main] - Qt Creator". The menu bar includes File, Edit, View, Build, Debug, Analyze, Tools, Window, and Help.

The Projects panel shows a single project named "NameHash [main]" with files "NameHash.pro" and "Sources". The "Sources" tab is selected, displaying the "NameHash.cpp" file content:

```
using namespace std;  
  
/* Prototype for the nameHash function. This lets us use the function  
 * 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); // Hash value is 0  
    cout << "The hash of your name is: " << hashValue << endl;  
}
```

A large yellow smiley face icon is overlaid on the left side of the code editor. A black callout bubble points from the bottom left towards the line of code where `hashValue` is assigned a value of 0. Inside the bubble, Chinese text asks, "但看起来我们在设置 hashValue 等于由 nameHash 函数返回的数字。这是怎么回事？" (But it looks like we're setting hashValue to the value returned by the nameHash function. What's going on?).

The status bar at the bottom shows tabs for Issues, Search Results, Application Output, Compile Output, QML Debugger Console, General Messages, Version Control, Test Results, and Views. The "Views" tab is currently selected.



但看起来我们在设置 `hashValue` 等于由函数返回的数字。这是怎么回事？

NameHash.cpp @ NameHash [main] - Qt Creator

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Debug Projects Help

Welcome Edit Design Debug Projects Help

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

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

```
using namespace std;  
/* Prototype for the nameHash function. This lets us use the function  
 * 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); // hashValue: 0  
  
    cout << "The hash of your name is: " << hashValue << endl;  
}
```

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

int hashValue = nameHash(first, last); hashValue: 0

int hashValue = nameHash(first, last); hashValue: 0

cout << "The hash of your name is: " << hashValue << endl;

cout << "The hash of your name is: " << hashValue << endl;

returned value 15058255 int

returned value 15058255 int

Name Value Type

Name Value Type

43 * treats each character in the input name as a number between 0 and 9.

43 * treats each character in the input name as a number between 0 and 9.

studentMain NameHash.cpp 31 0x5555555ab0fb
std::Function_Invoker<int>::operator()<int>(void*) 0x55555556037fc
GThreadStd::run() 0x5555555e6616
?? 0x7ffff64dc2b3
start_thread pthread_create.c 442 0x7ffff6094b43
clone3 clone3.S 81 0x7ffff6126a00

studentMain NameHash.cpp 31 0x5555555ab0fb
std::Function_Invoker<int>::operator()<int>(void*) 0x55555556037fc
GThreadStd::run() 0x5555555e6616
?? 0x7ffff64dc2b3
start_thread pthread_create.c 442 0x7ffff6094b43
clone3 clone3.S 81 0x7ffff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results

The screenshot shows the Qt Creator IDE interface with the following details:

- File Menu:** File, Edit, View, Build, Debug, Analyze, Tools, Window, Help.
- Projects:** NameHash [main] (selected), NameHash.pro, Sources, NameHash.cpp.
- Code Editor:** The main window displays the `NameHash.cpp` file content:

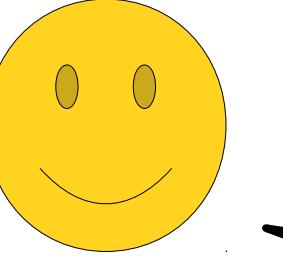
```
using namespace std;
/* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.
*/
int nameHash(string first, string last);

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

    int hashValue = nameHash(first, last); // Line 31

    cout << "The hash of your name is: " << hashValue << endl;
}
```
- Variables View:** Shows local variables at line 31, col 5:

Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string
- Output View:** Shows the returned value: `15058255 int`.
- Callout Box:** A large black callout box contains handwritten text: "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 View:** Shows the stack trace for the stopped thread #15 NameHash.
- Bottom Bar:** Type to locate (Ctrl...), Issues, Search Results, Application Output, Compile Output, QML Debugger Console, General Messages, Version Control, Test Results.



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!

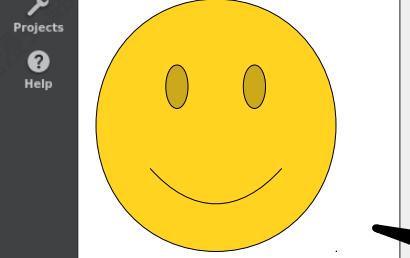
The screenshot shows the Qt Creator IDE interface with a yellow smiley face icon in the top-left corner. A large callout box with a black border and rounded corners is overlaid on the code editor area. Inside the callout box, the following text is written:

发生的事情是我们刚刚返回
从 nameHash 带有一个值，但由于我们正在
一步一步地通过程序，
我们实际上还没有将那个值分配给
hashValue yet!

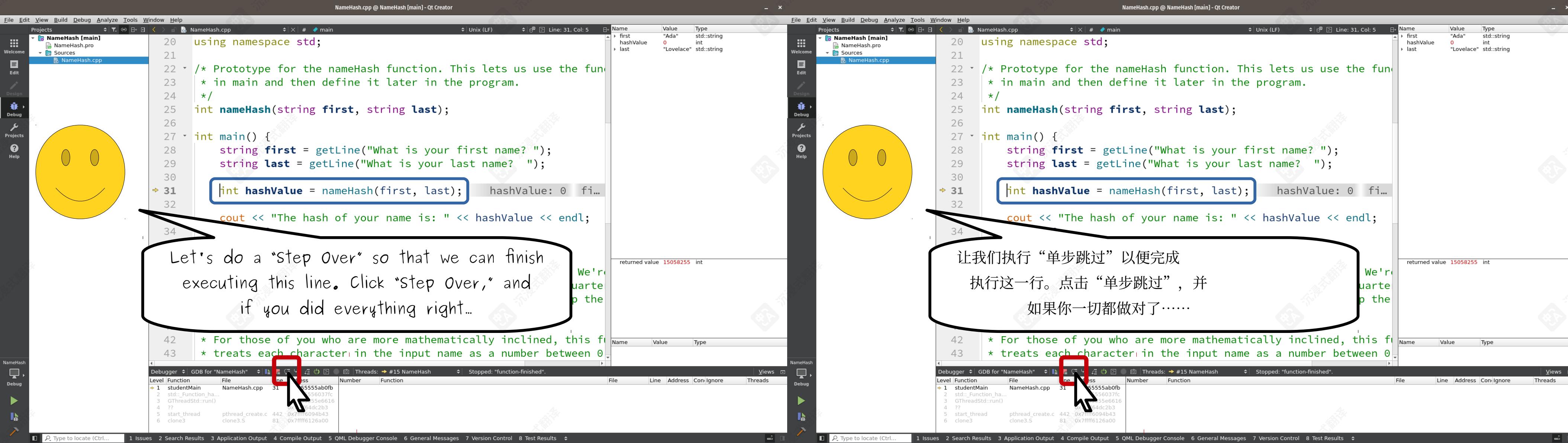
The code editor displays C++ code for calculating a hash value from two names. The variable `hashValue` is highlighted with a blue rectangle. The status bar at the bottom right shows the returned value as 15058255, which is an int.

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

Level	Function	File	Line	Address	Threads	Stopped
1	studentMain	NameHash.cpp	31	0x5555555ab0fb	#15 NameHash	#15 NameHash
2	std::function<_T...>::operator()<_T...>			0x5555556037fc		
3	GThreadStd::run()			0x5555555e6616		
4	??			0x7ffff64dc2b3		
5	start_thread	pthread_create.c	442	0x7ffff6094b43		
6	clone3	clone3.S	81	0x7ffff6126a00		



发生的事情是我们刚刚返回
从 `nameHash` 带有一个值，但由于
一步一步地通过程序，
我们实际上还没有将那个值分配给
`hashValue` 变量。



NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

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

```
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets us use the function
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); // hashValue: 0
32
33     cout << "The hash of your name is: " << hashValue << endl;
34 }
```

int hashValue = nameHash(first, last); // hashValue: 0

cout << "The hash of your name is: " << hashValue << endl;

returned value 15058255 int

Name Value Type

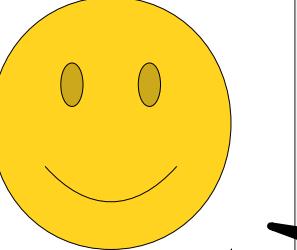
Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "function-finished".

Level Function File Address Con Ignore Threads

Level	Function	File	Address	Con	Ignore	Threads
1	studentMain	NameHash.cpp	31	0x5555ab0fb		
2	std::Function_h...			0x556037fc		
3	GThreadStd::run()			0x55e6616		
4	???			0x7ffff64dc2b3		
5	start_thread	pthread_create.c	442	0x7fff6094b43		
6	clone3	clone3.S	81	0x7fff6126a00		

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results



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

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

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

```
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets us use the function
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); // hashValue: 0
32
33     cout << "The hash of your name is: " << hashValue << endl;
34 }
```

int hashValue = nameHash(first, last); // hashValue: 0

cout << "The hash of your name is: " << hashValue << endl;

returned value 15058255 int

Name Value Type

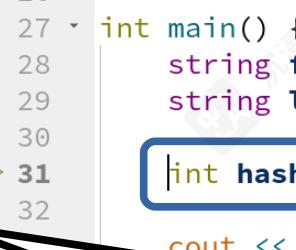
Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "function-finished".

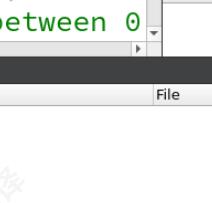
Level Function File Address Con Ignore Threads

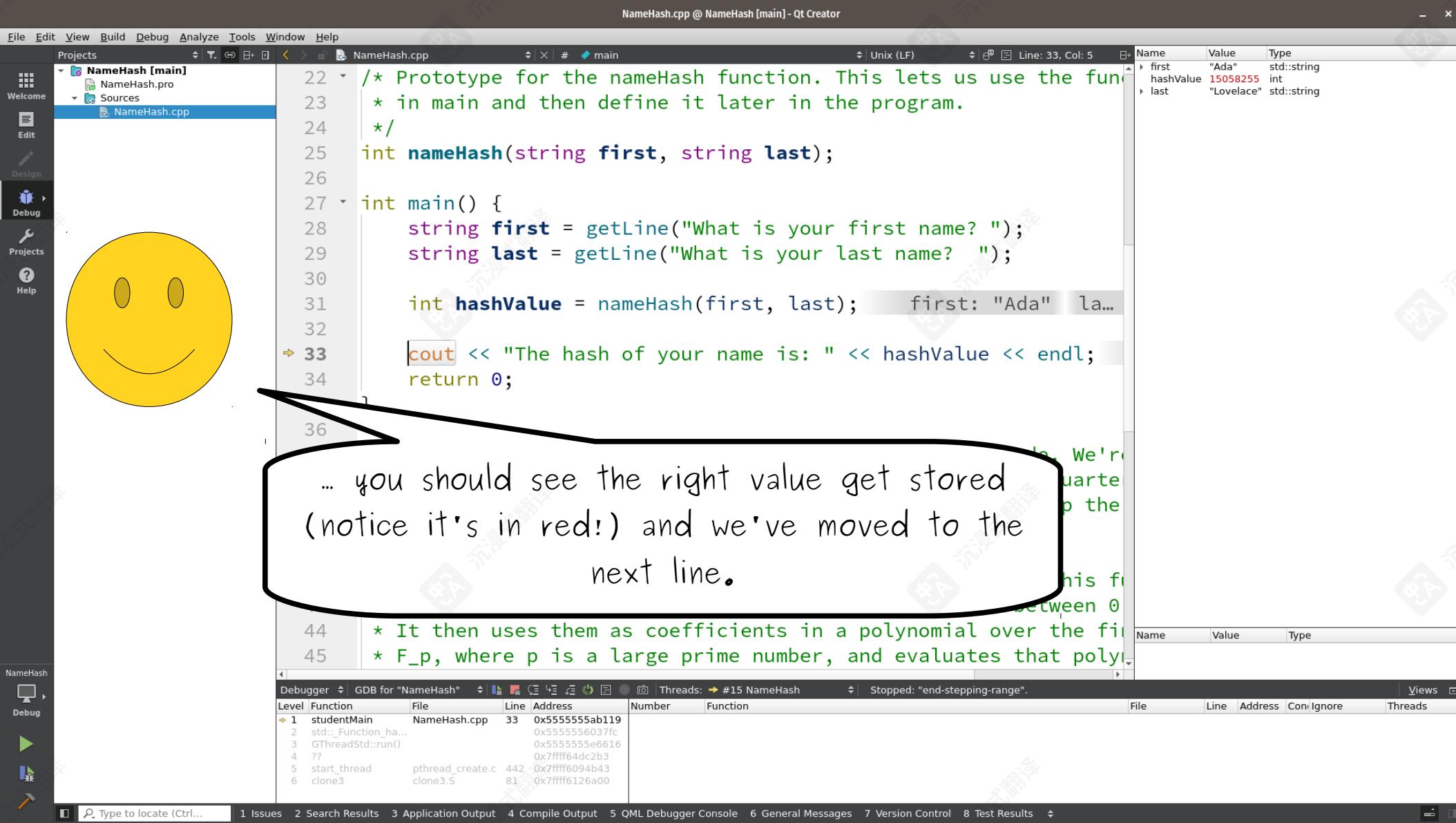
Level	Function	File	Address	Con	Ignore	Threads
1	studentMain	NameHash.cpp	31	0x5555ab0fb		
2	std::Function_h...			0x556037fc		
3	GThreadStd::run()			0x55e6616		
4	???			0x7ffff64dc2b3		
5	start_thread	pthread_create.c	442	0x7fff6094b43		
6	clone3	clone3.S	81	0x7fff6126a00		

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results



让我们执行“单步跳过”以便完成执行这一行。点击“单步跳过”，并如果你一切都做对了……





/* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.

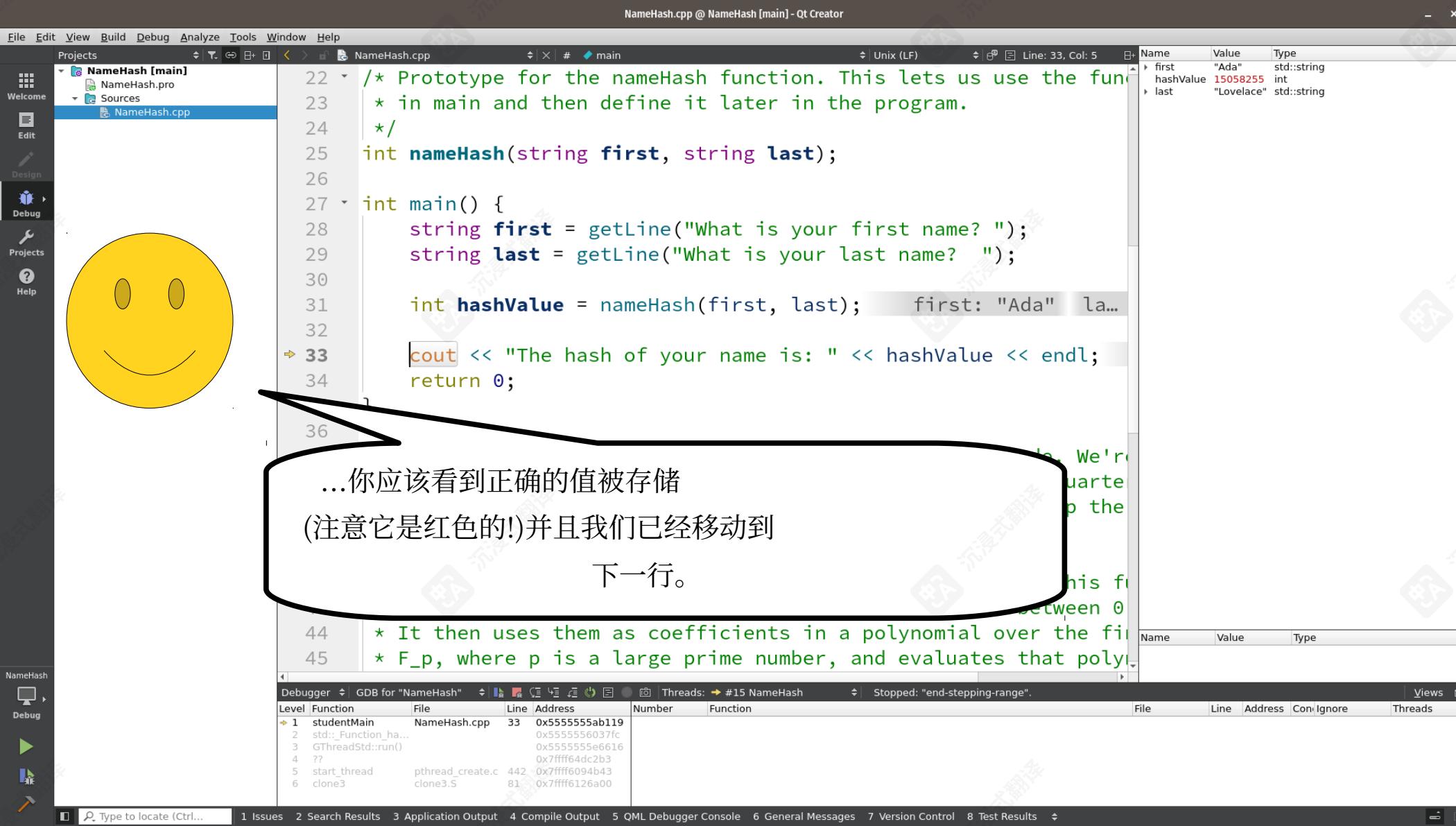
```
int nameHash(string first, string last);
```

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

 int hashValue = nameHash(first, last); first: "Ada" la...

 cout << "The hash of your name is: " << hashValue << endl;
 return 0;

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



/* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.

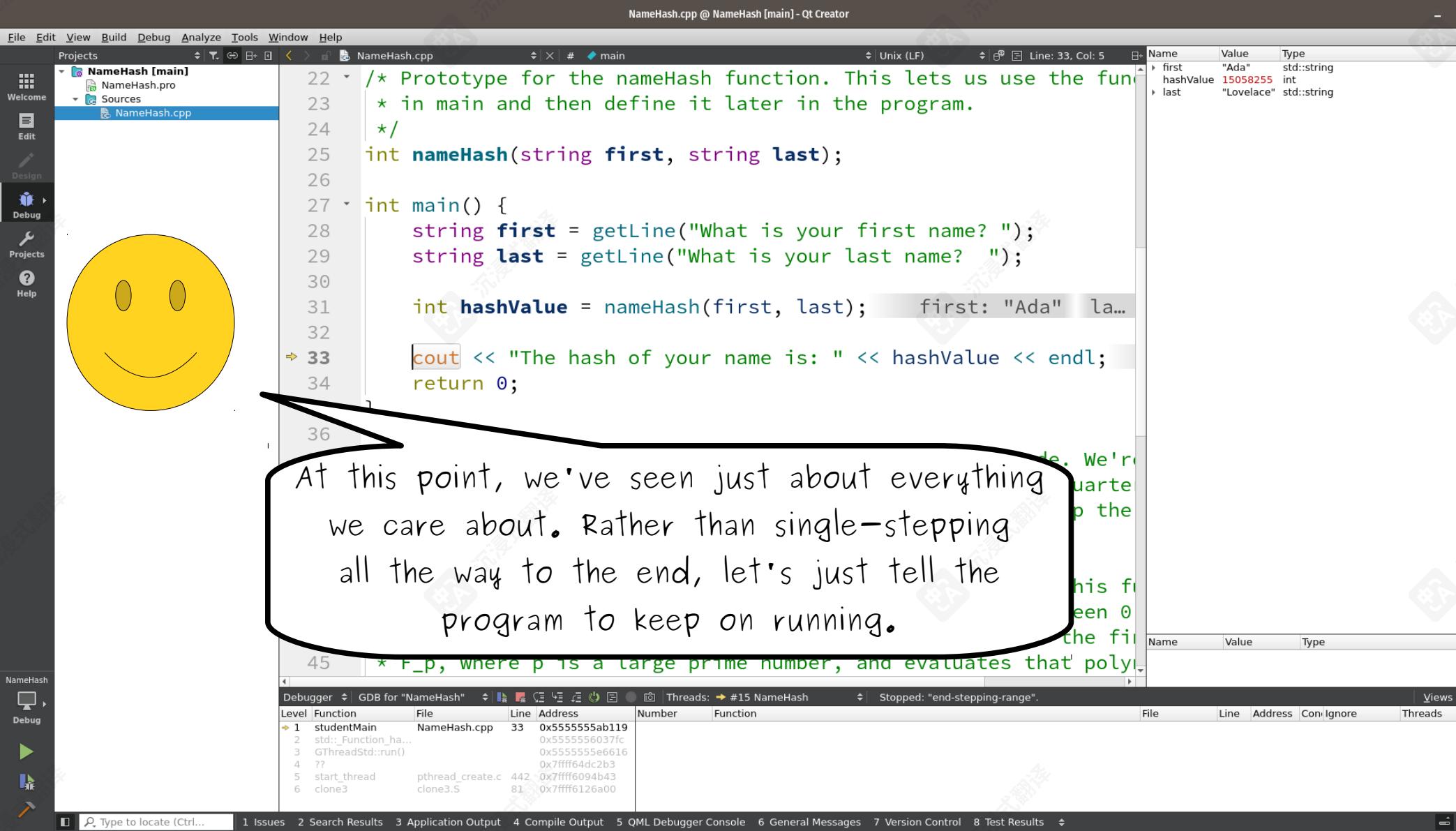
```
int nameHash(string first, string last);
```

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

 int hashValue = nameHash(first, last); first: "Ada" la...

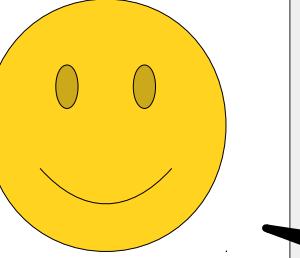
 cout << "The hash of your name is: " << hashValue << endl;
 return 0;

...你应该看到正确的值被存储
(注意它是红色的!)并且我们已经移动到
下一行。



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.

```
22  /* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.
23  */
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);    first: "Ada"  la...
32
33      cout << "The hash of your name is: " << hashValue << endl;
34      return 0;
35
36
45  * F_p, where p is a large prime number, and evaluates that polynomial.
```



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.

The screenshot shows the Qt Creator IDE interface. On the left, there's a vertical toolbar with icons for Welcome, Edit, Design, Debug, Projects, and Help. The main area has a dark theme with a light gray sidebar. A yellow smiley face icon is positioned on the left side of the code editor. A large black callout box highlights the following text in the code editor:

```
/* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.
```

The code editor shows a file named NameHash.cpp with the following content:

```
22  /* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.
```

```
23  */


```
24 int nameHash(string first, string last);
```

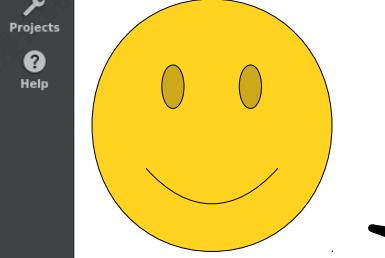


```
25
26
27 int main() {
28 string first = getLine("What is your first name? ");
29 string last = getLine("What is your last name? ");
30
31 int hashValue = nameHash(first, last); first: "Ada" la...
32
33 cout << "The hash of your name is: " << hashValue << endl;
34 return 0;
35
36
37
38
39
40
41
42
43
44
45 * F_p, where p is a large prime number, and evaluates that polynomial.
```



A status bar at the bottom displays various toolbars and tabs. A small callout box contains the Chinese text: “此时，我们已经看到了所有我们关心的事情。与其一步步走到最后，不如告诉程序继续运行。” (At this point, we have seen all the things we care about. Instead of walking step by step to the end, let's tell the program to continue running.)


```



此时，我们已经看到了所有我们关心的事情。与其一步步走到最后，不如告诉程序继续运行。

File Edit View Build Debug Analyze Tools Window Help

NameHash.cpp @ NameHash [main] - Qt Creator

Projects NameHash [main] Sources NameHash.cpp

```
22  /* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.
```

```
23  */


```
24 int nameHash(string first, string last);
```



```
25
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); first: "Ada" la...
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 code.
38 * to 'talk'
39 * the
40 * of
41 *
42 * For
43 * tre...
44 * It
45 * F_p,
```



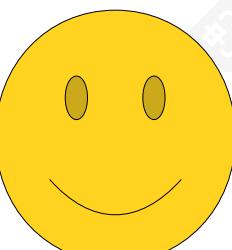
Line: 33, Col: 5



| Name      | Value      | Type        |
|-----------|------------|-------------|
| first     | "Ada"      | std::string |
| hashValue | 15058255   | int         |
| last      | "Lovelace" | std::string |



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."



Debugger GDB for "NameHash" Threads: #15 NameHash Stopped: "end-stepping-range".



| Level | Function           | File             | Line | Address         | Number | Function | File | Line | Address | Con | Ignore | Threads |
|-------|--------------------|------------------|------|-----------------|--------|----------|------|------|---------|-----|--------|---------|
| 1     | studentMain        | NameHash.cpp     | 33   | 0x5555555ab119  |        |          |      |      |         |     |        |         |
| 2     | std::function<...> |                  |      | 0x55555556037fc |        |          |      |      |         |     |        |         |
| 3     | GThreadStd::run()  |                  |      | 0x55555555e6616 |        |          |      |      |         |     |        |         |
| 4     | ??                 |                  |      | 0x7ffff64dc2b3  |        |          |      |      |         |     |        |         |
| 5     | start_thread       | pthread_create.c | 442  | 0x7ffff6094b43  |        |          |      |      |         |     |        |         |
| 6     | clone3             | clone3.S         | 81   | 0x7ffff6126a00  |        |          |      |      |         |     |        |         |



Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 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."

File Edit View Build Debug Analyze Tools Window Help

Welcome Edit Design Projects Projects NameHash [main] NameHash.pro Sources NameHash.cpp

NameHash.cpp @ NameHash [main] - Qt Creator

Line: 33, Col: 5

```
22  /* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.
```

```
23  */
```

```
24  int nameHash(string first, string last);
```

```
25
```

```
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);    first: "Ada"  la...
```

```
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 code.
```

```
38  * to talk to the user about the computation.
```

```
39  * For example, if the user asks for the hash of "Ada Lovelace",
```

```
40  * the program would respond with "The hash of Ada Lovelace is 15058255".
```

```
41  *
```

```
42  * For now, we'll just return a random value.
```

```
43  *
```

```
44  * It's important to note that this is a very simple implementation.
```

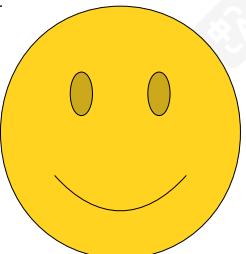
```
45  *
```

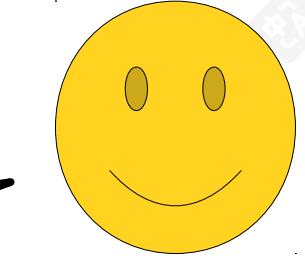
要执行此操作，请点击此按钮。如果你悬停在它上面，它显示“继续”，并且那个按钮表示“暂停程序并让它从这里继续运行。”

ver
n
ep

1 studentMain NameHash.cpp 33 0x5555555ab119
2 std::function<...> 0x5555556037fc
3 GThreadStd::run() 0x5555555e6616
4 ?? 0x7ffff64dc2b3
5 start_thread pthread_create.c 442 0x7ffff6094b43
6 clone3 clone3.S 81 0x7ffff6126a00

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results





操作，请点击此按钮。如果你悬停
面，它显示“继续”，并且那个按
“暂停程序并让它从这
里继续运行”

File Edit View Build Debug Analyze Tools Window Help

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

```
22  /* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.
```

```
23  */
24  int nameHash(string first, string last);
25
26
27  int main() {
28      NameHash Console[Completed]
29      File Edit Options Help
30      A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
31      What is your first name? Ada
32      What is your last name? Lovelace
33      The hash of your name is: 15058255
34
35
36
37
38
39
40
41
42
43
44 * It then
45 * F_p, where p is a large prime number, and evaluates that polynomial
```

If you do, you should see something like this.
(The program window might not automatically pop up. That's okay! Just open it manually.)

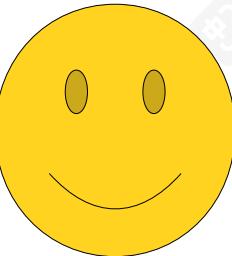
Our program is now done running!

Value Type

Debugger GDB for "NameHash" Threads: #14 NameHash:sh5 Running.

Level	Function	File	Line	Address	Number	Function
1	studentMain	NameHash.cpp	33	0x5555555ab119		
2	std::function<...>			0x5555556037fc		
3	GThreadStd::run()			0x5555555e6616		
4	??			0x7ffff64dc2b3		
5	start_thread	pthread_create.c	442	0x7ffff094b43		
6	clone3	clone3.S	81	0x7ffff6126a00		

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 7 Version Control 8 Test Results





you do, you should see something like this.
The program window might not automatically
open up. That's okay! Just open it manually.)
Our program is now done running!

gram is now done running!

The screenshot shows the Qt Creator IDE interface. The top menu bar includes File, Edit, View, Build, Debug, Analyze, Tools, Window, and Help. The left sidebar has icons for Welcome, Edit, Design, Debug, Projects, and Help. The central area displays the code for NameHash.cpp:

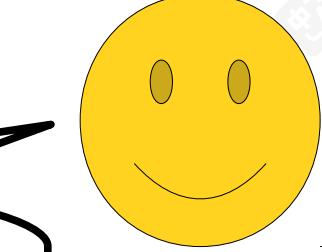
```
22  /* Prototype for the nameHash function. This lets us use the function in main and then define it later in the program.
23  */
24
25  int nameHash(string first, string last);
26
27  int main() {
28      // ...
29      cout << "What is your first name? " << first;
30      cout << "What is your last name? " << last;
31      cout << "The hash of your name is: " << hash(first + last);
32      cout << endl;
33  }
```

A terminal window titled "NameHash Console [Completed]" is open, showing the output of the program:

```
What is your first name? Ada
What is your last name? Lovelace
The hash of your name is: 15058255
```

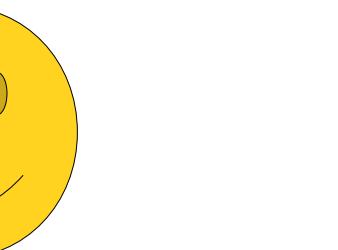
To the right of the terminal, there is a large yellow smiley face icon. A black callout bubble points from the text in the terminal towards the smiley face. The text in the bubble is:

如果你这样做，你应该能看到类似这样的事情。
(程序窗口可能不会自动
弹出。没关系！手动打开它就行了。)
我们的程序现在运行完成了！



如果你这样做，你应该能看到类似
(程序窗口可能不会自动
弹出。没关系！手动打开它就行)
我们的程序现在运行完成

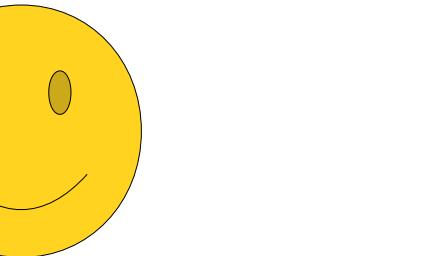
So there you have it! You've now gotten more familiar with the debugger!



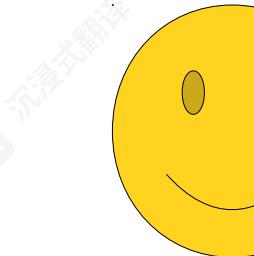
就这样！你现在对调试器更加熟悉了！



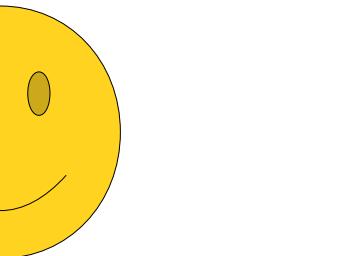
You know how to set a breakpoint to pause the program at a particular point.



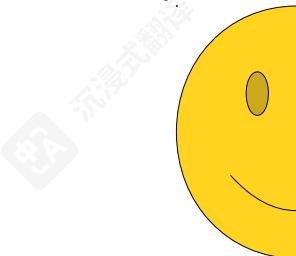
你知道如何设置断点来在特定位置暂停程序。



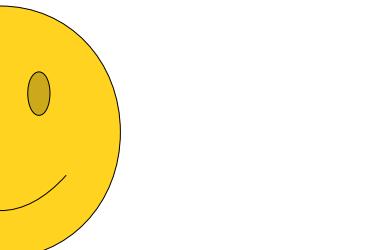
You know how to read the call stack and to see the values of local variables.



你知道如何阅读调用堆栈并查看局部变量的值。



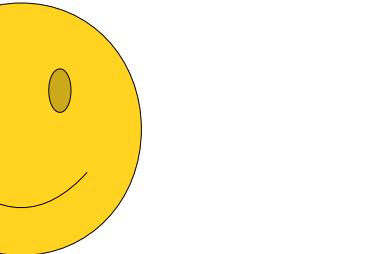
You know how to single-step the program and see what values change.



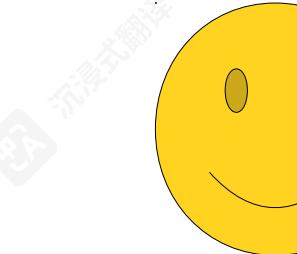
你知道如何单步执行程序并查看哪些值发生了变化。



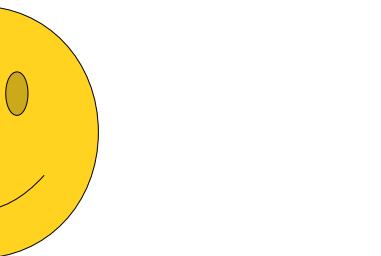
You know how to run a function to completion,
and how to let the program keep on running.



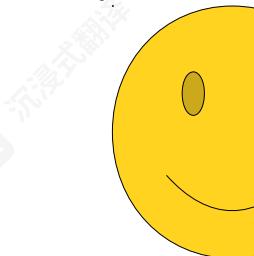
你知道如何运行一个函数以完成，以及如何让程序继续运行。



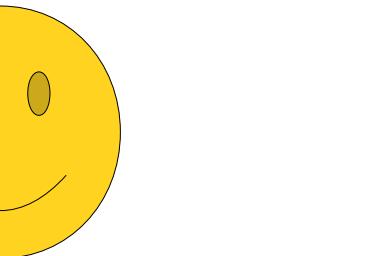
As you write more and more complicated programs this quarter, you'll get a lot more familiar using the debugger and seeing how your programs work.



随着这个季度你编写越来越复杂的程序，你会越来越熟悉使用调试器并了解你的程序是如何工作的。



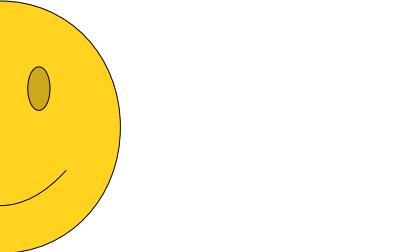
And, if you continue to build larger and larger pieces of software, you'll find that knowing how to use a debugger is a surprisingly valuable skill!



而且，如果你继续构建越来越大的软件，你会发现知道如何使用调试器是一项非常宝贵的技能！



Hope this helps, and welcome to CS106B!



希望这有帮助，欢迎来到CS106B！

