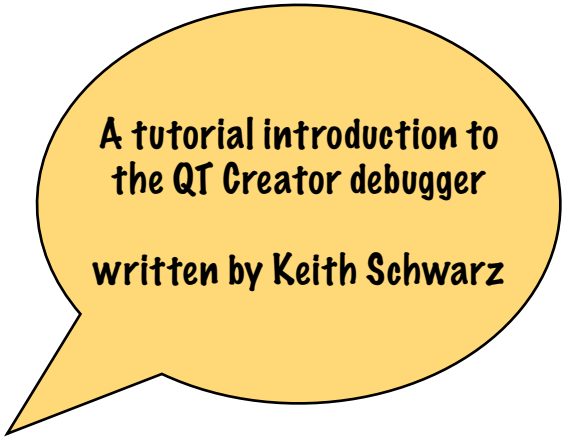
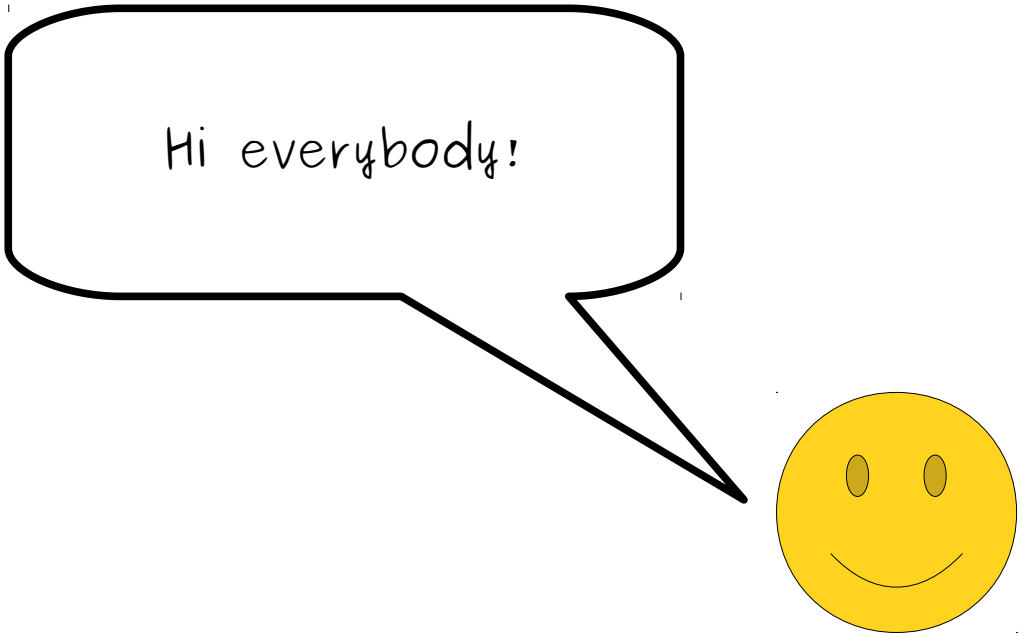
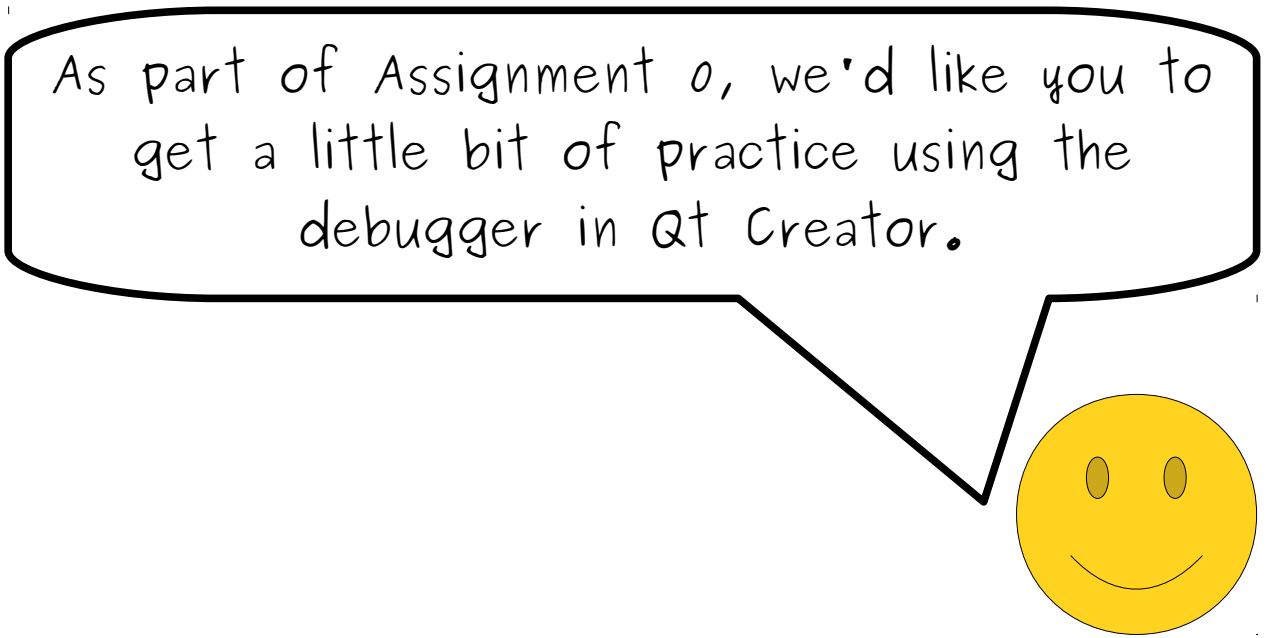
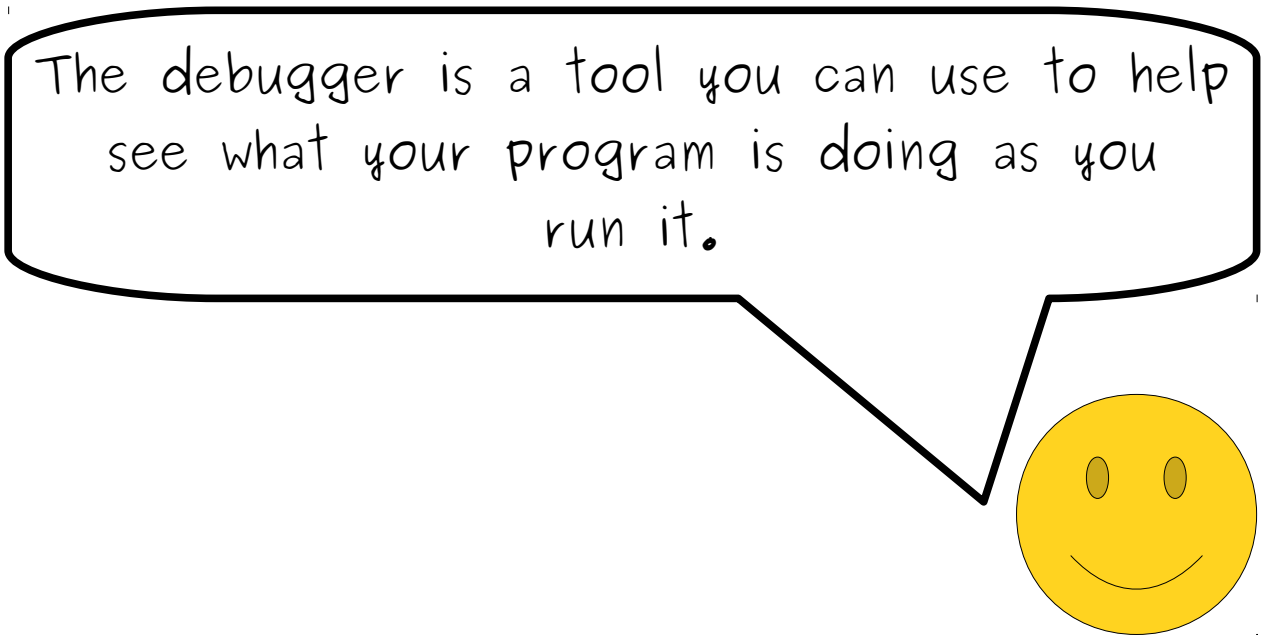
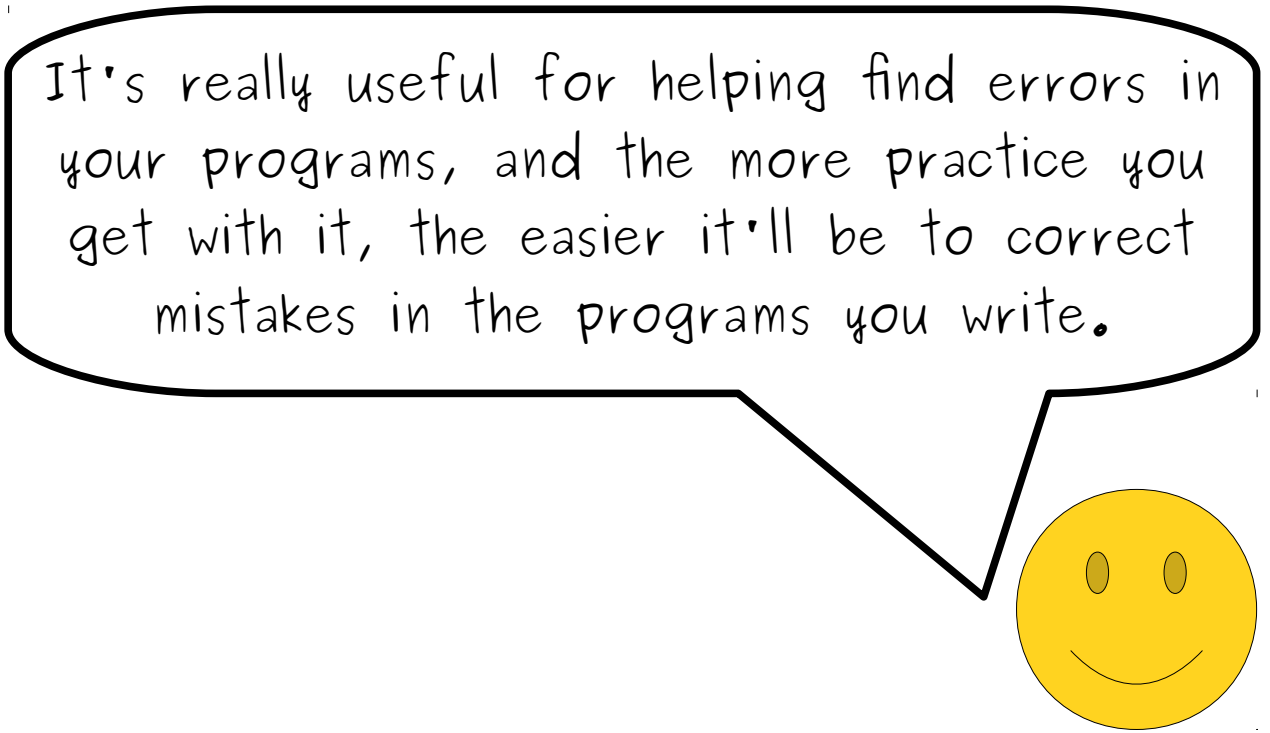


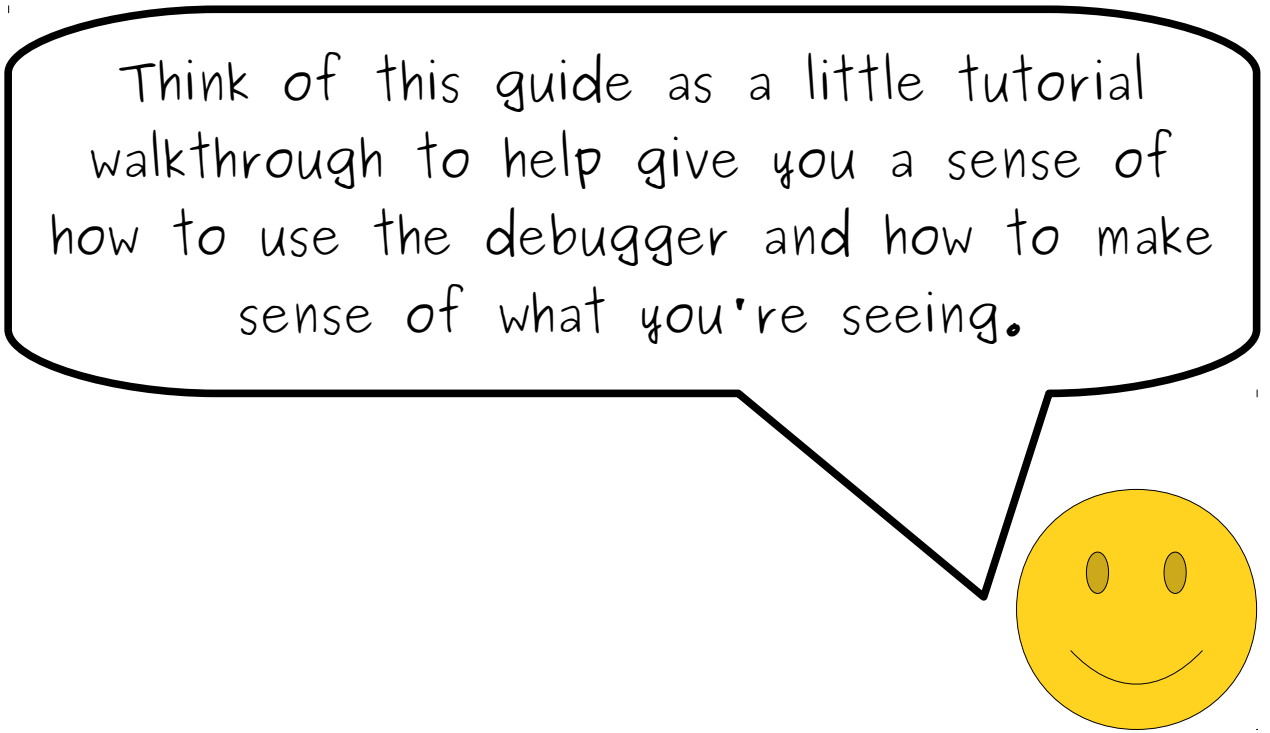
Assignment 0: Using the Debugger


Hi everybody!

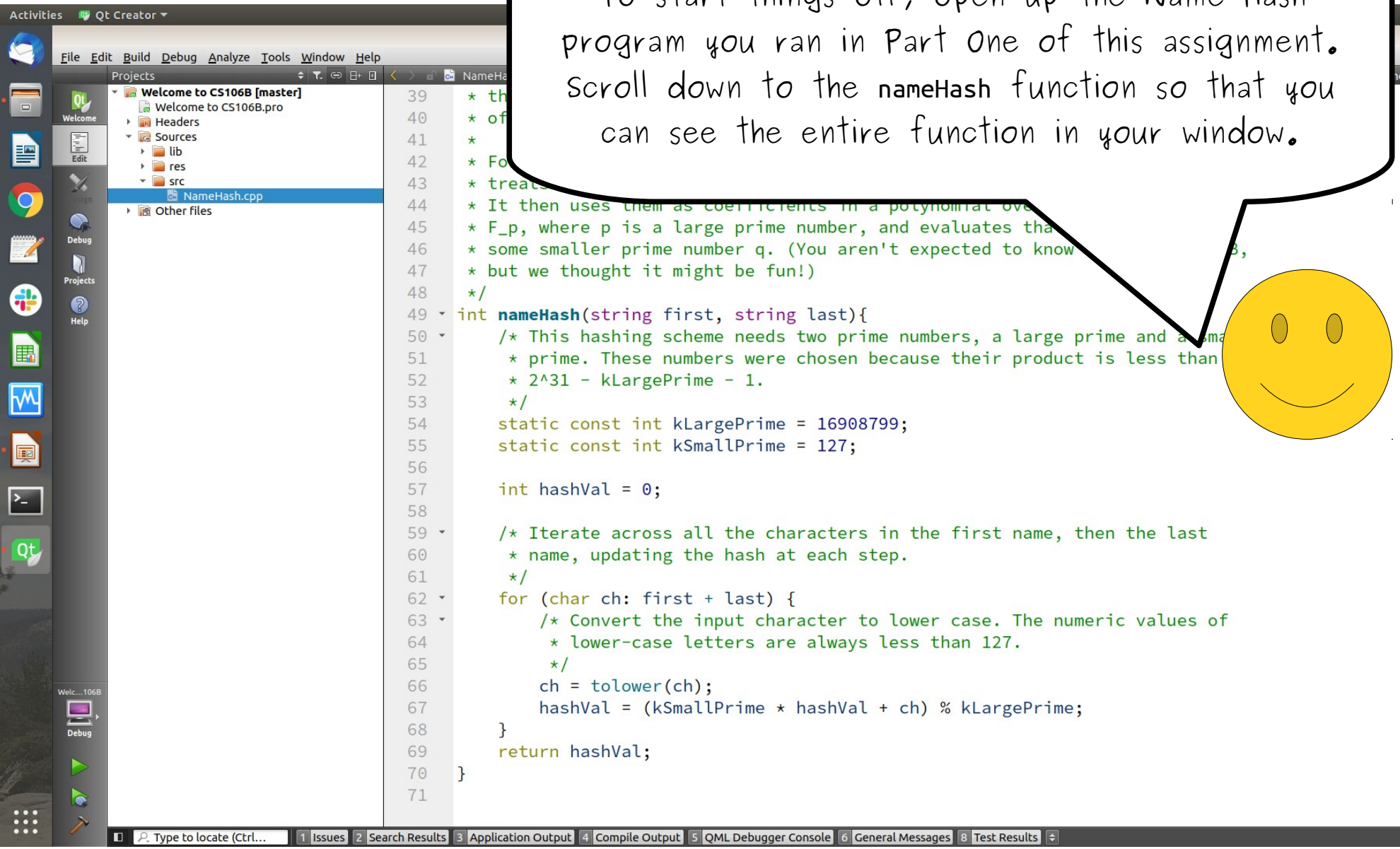

As part of Assignment 0, we'd like you to get a little bit of practice using the debugger in Qt Creator.

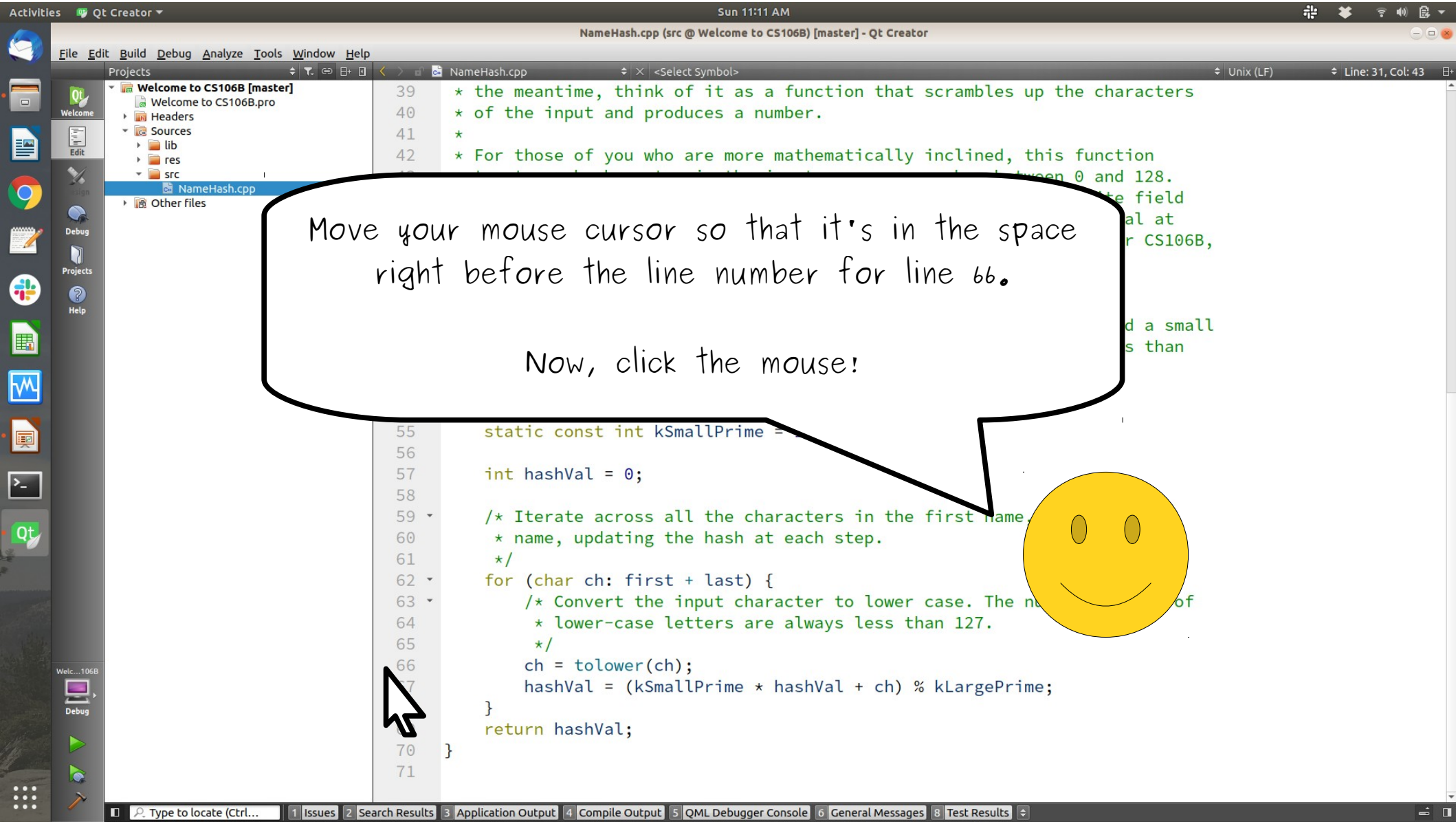

The debugger is a tool you can use to help see what your program is doing as you run it.

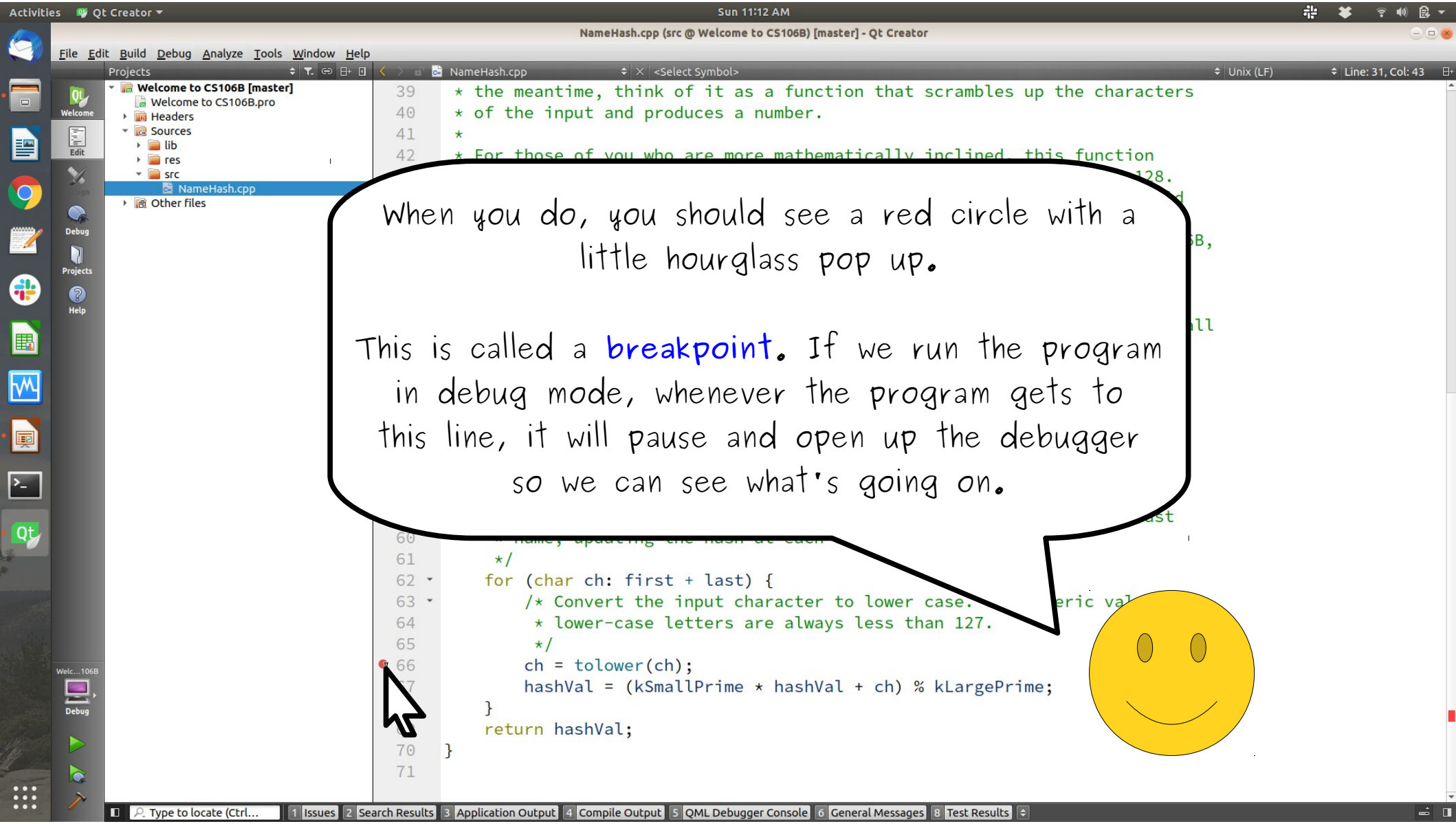

It's really useful for helping find errors in your programs, and the more practice you get with it, the easier it'll be to correct mistakes in the programs you write.

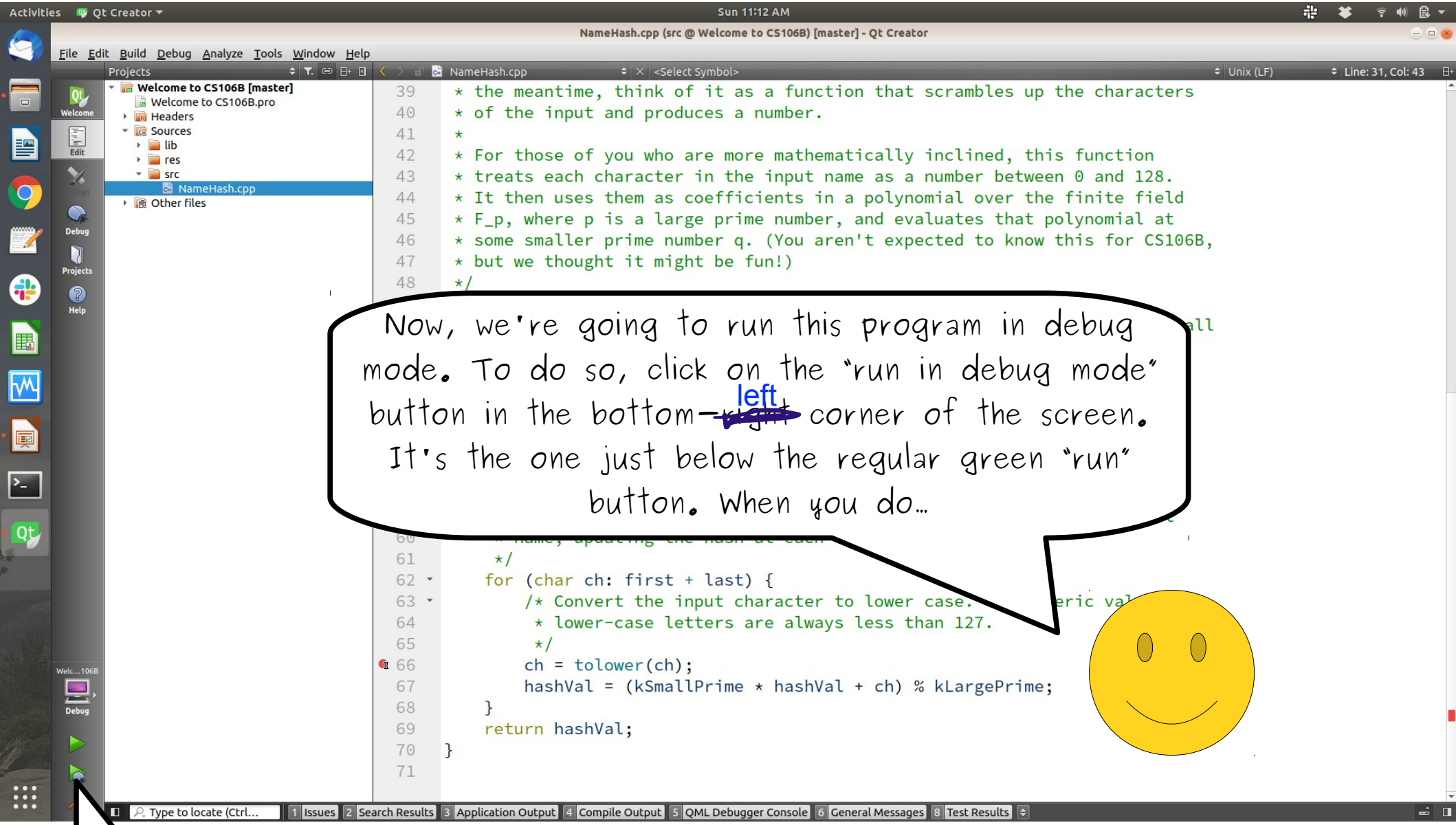

Think of this guide as a little tutorial walkthrough to help give you a sense of how to use the debugger and how to make sense of what you're seeing.

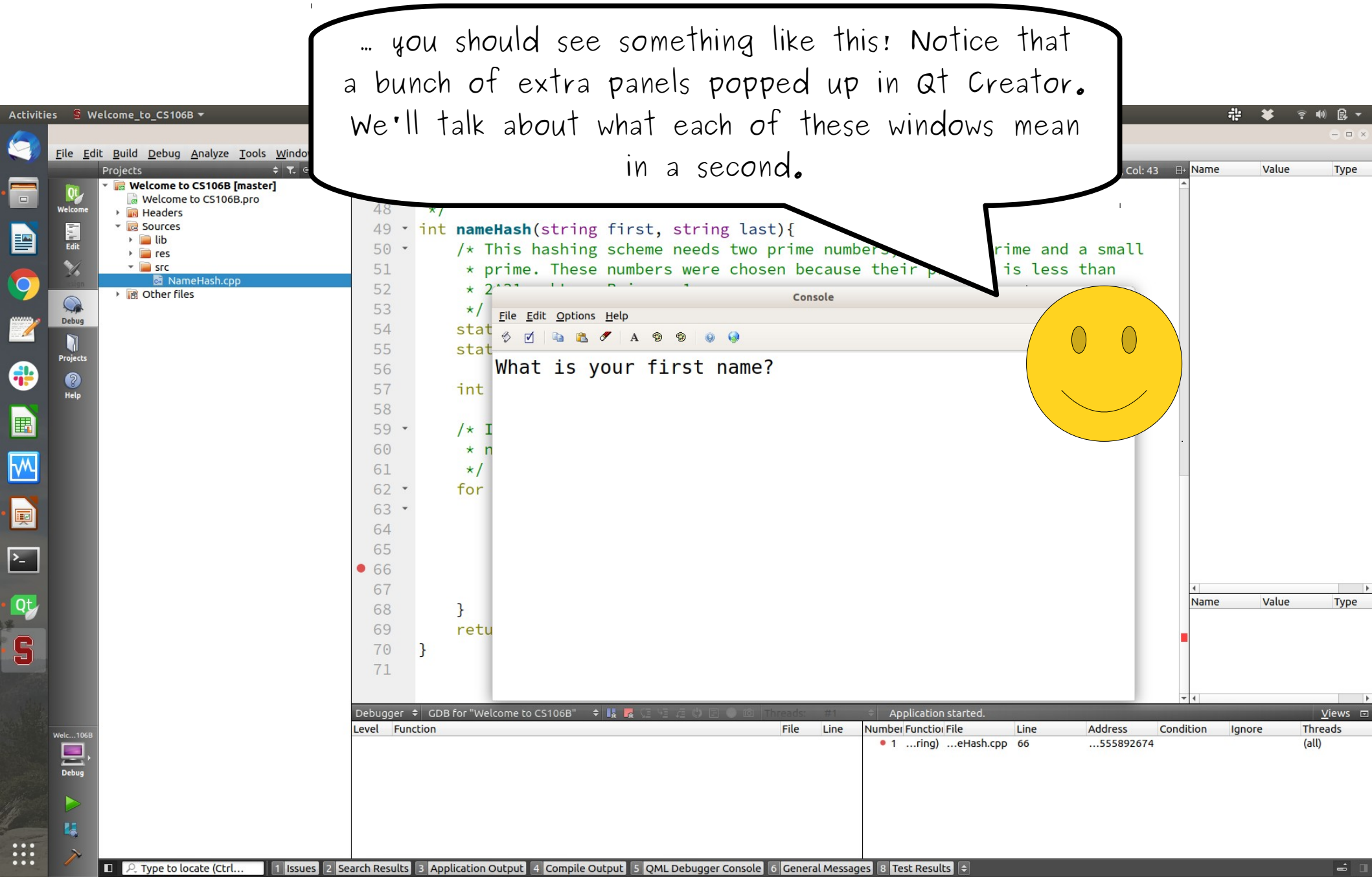
To start things off, open up the Name Hash program you ran in Part One of this assignment. Scroll down to the **nameHash** function so that you can see the entire function in your window.

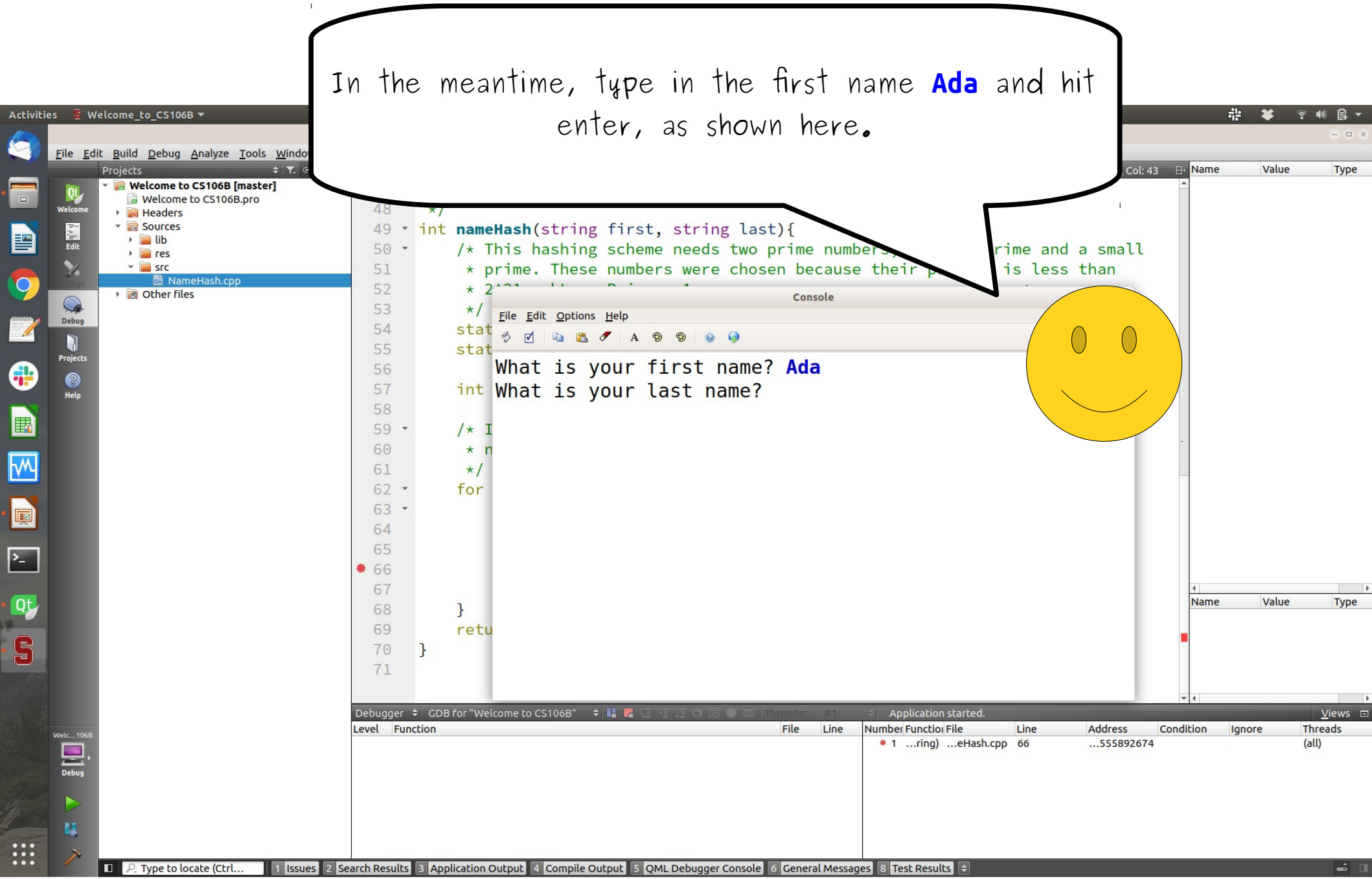



Activities
Qt Creator
Sun 11:11 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 43
* the meantime, think of it as a function that scrambles up the characters
* of the input and produces a number.
*
* For those of you who are more mathematically inclined, this function
Move your mouse cursor so that it's in the space right before the line number for line 66.
Now, click the mouse!
static const int kSmallPrime =
int hashVal = 0;
/* Iterate across all the characters in the first name,
* name, updating the hash at each step.
*/
for (char ch: first + last) {
/* Convert the input character to lower case. The nu of
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

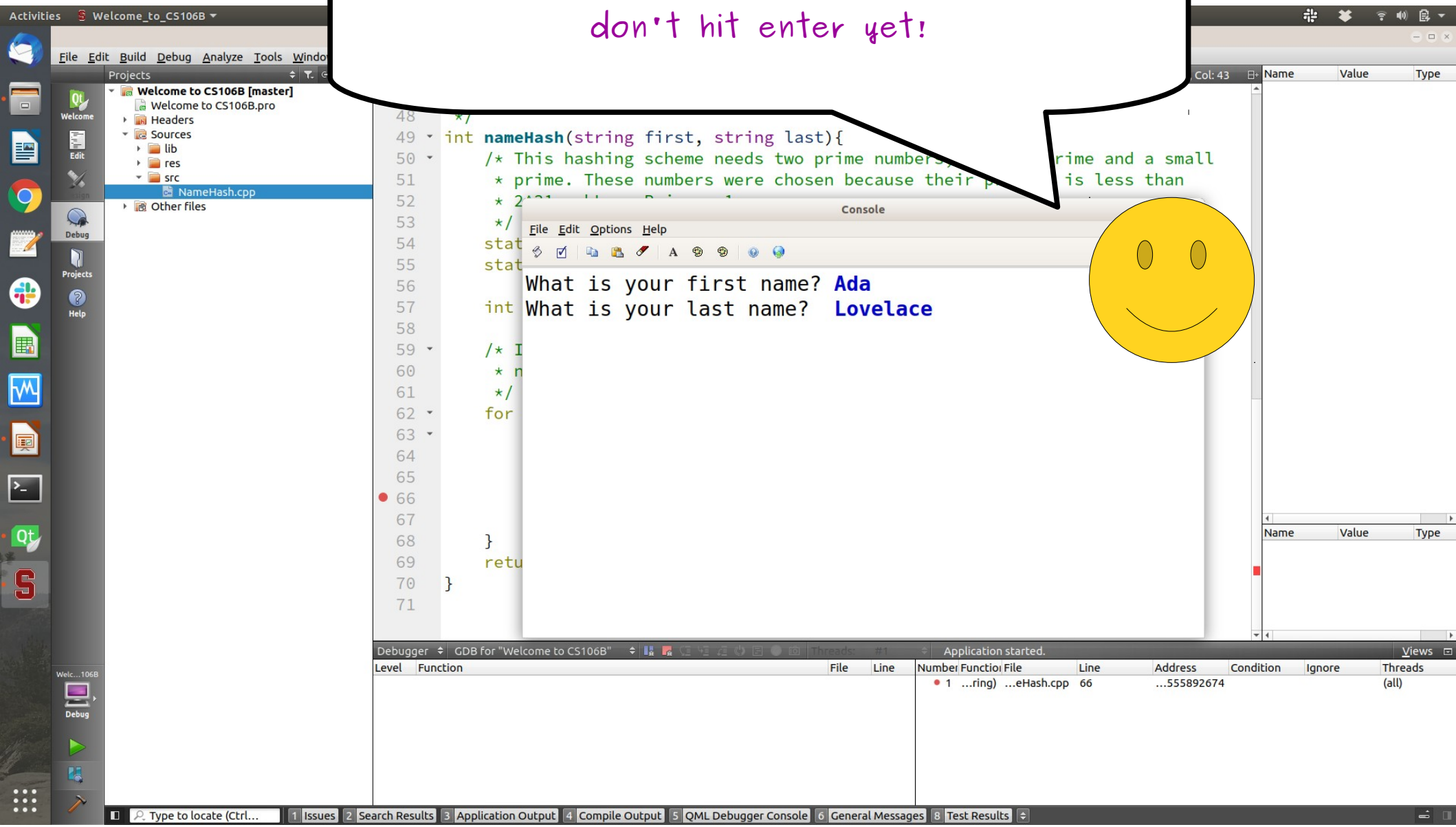

Activities
Qt Creator
Sun 11:12 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 43
* the meantime, think of it as a function that scrambles up the characters
* of the input and produces a number.
*
* For those of you who are more mathematically inclined, this function
*/
for (char ch: first + last) {
/* Convert the input character to lower case.
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
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.
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results


Activities
Qt Creator
Sun 11:12 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 43
* the meantime, think of it as a function that scrambles up the characters
* 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 128.
* It then uses them as coefficients in a polynomial over the finite field
* F_p, where p is a large prime number, and evaluates that polynomial at
* some smaller prime number q. (You aren't expected to know this for CS106B,
* but we thought it might be fun!)
*/
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-left corner of the screen. It's the one just below the regular green "run" button. When you do…
left
*/
for (char ch: first + last) {
/* Convert the input character to lower case.
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Welc...106B
Debug
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

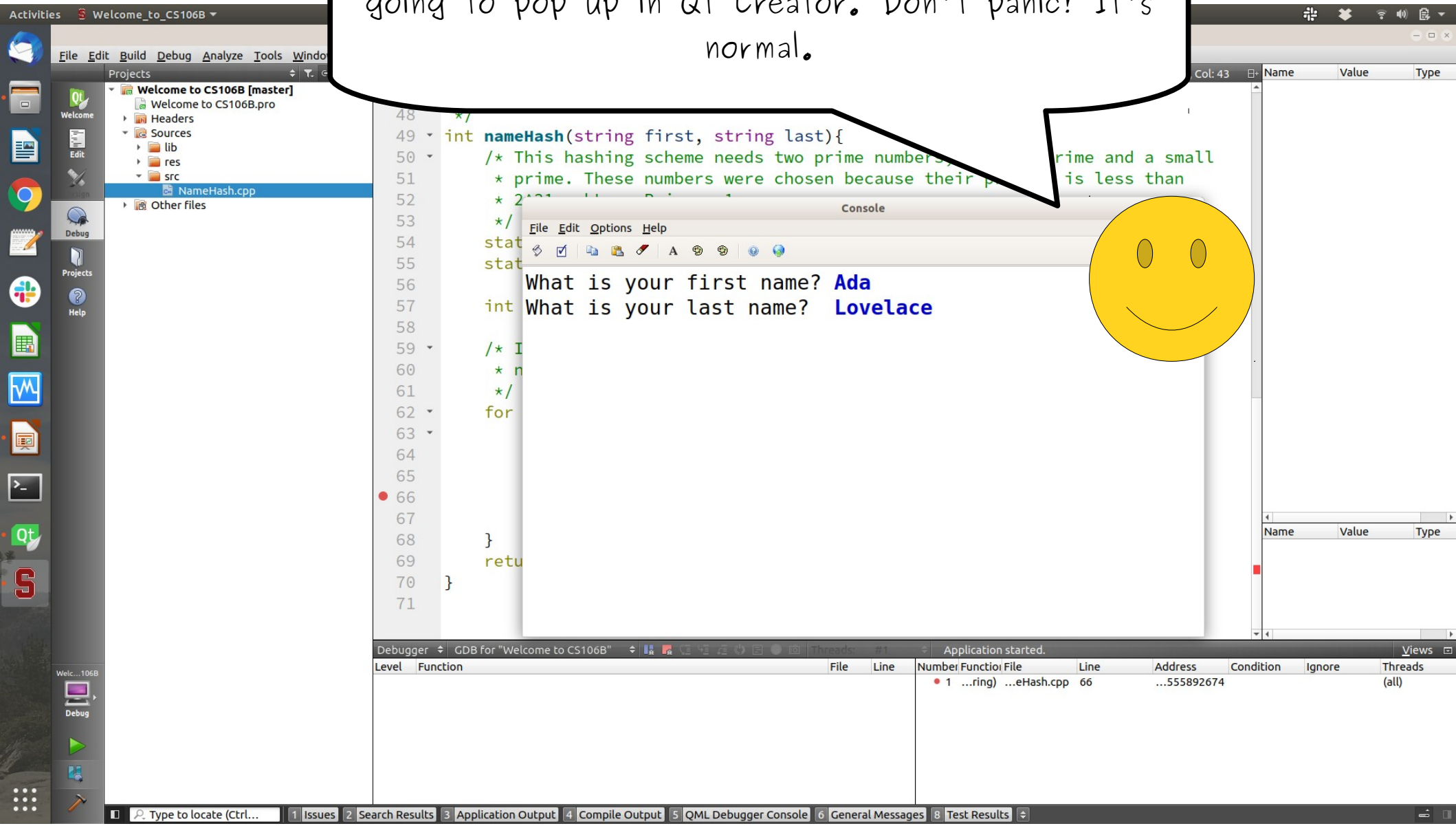

… you should see something like this! Notice that a bunch of extra panels popped up in Qt Creator. We'll talk about what each of these windows mean in a second.
Activities
Welcome_to_CS106B
File Edit Build Debug Analyze Tools Windo
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
Welcome
Edit
Debug
Projects
Help
Col: 43
Name Value Type
int nameHash(string first, string last){
/* This hashing scheme needs two prime numbers,
* prime. These numbers were chosen because their
is less than
Console
File Edit Options Help
What is your first name?
Debugger
GDB for "Welcome to CS106B"
Application started.
Views
Level Function File Line
Number Function File Line Address Condition Ignore Threads
1 …ring) …eHash.cpp 66 …555892674 (all)
Type to locate (Ctrl…
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

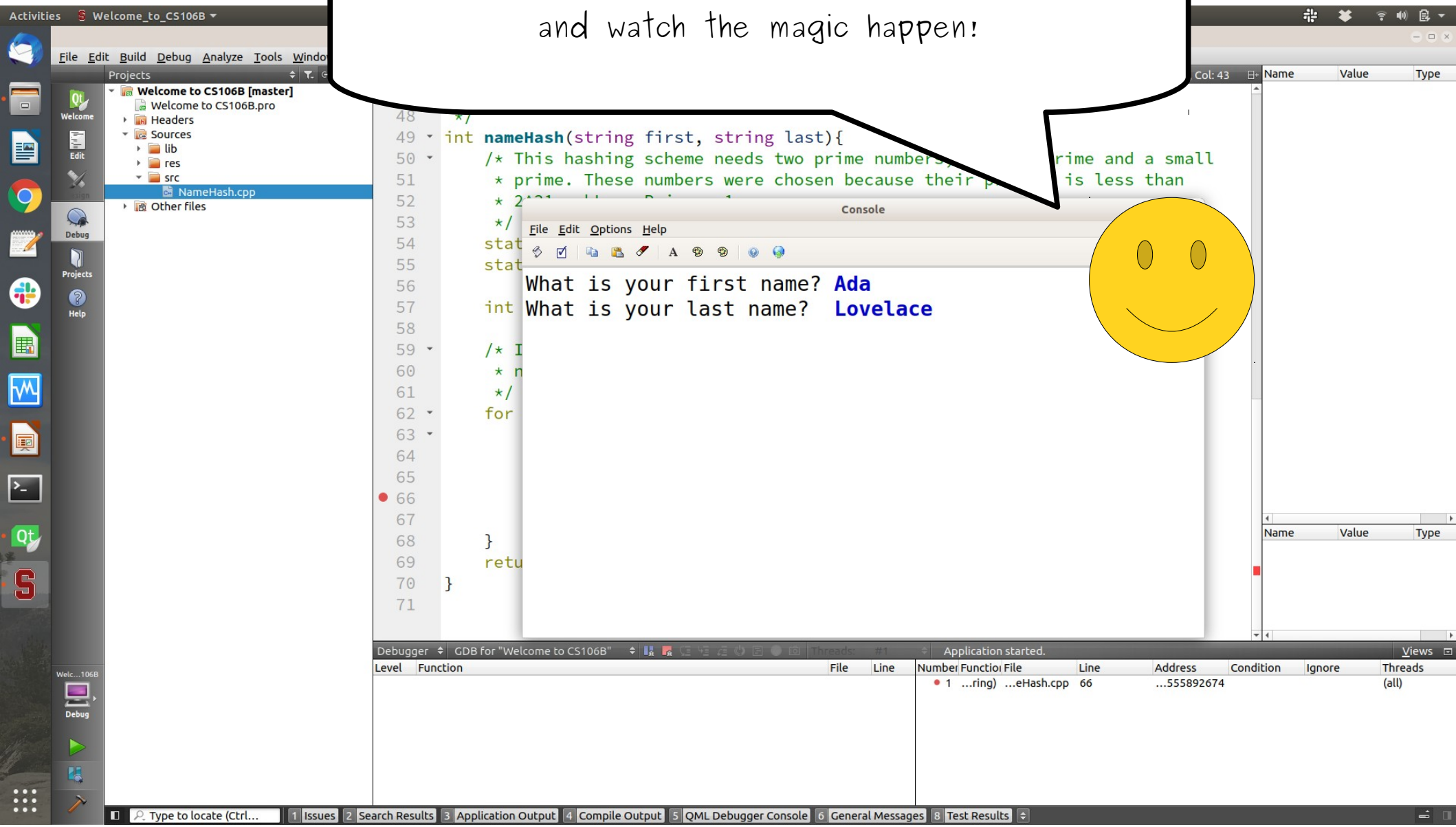

In the meantime, type in the first name Ada and hit enter, as shown here.
Activities
Welcome_to_CS106B
File Edit Build Debug Analyze Tools
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
Welcome
Edit
Debug
Projects
Help
Console
File Edit Options Help
What is your first name? Ada
What is your last name?
Debugger
GDB for "Welcome to CS106B"
Application started.
Views
Level Function File Line
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Name Value Type
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

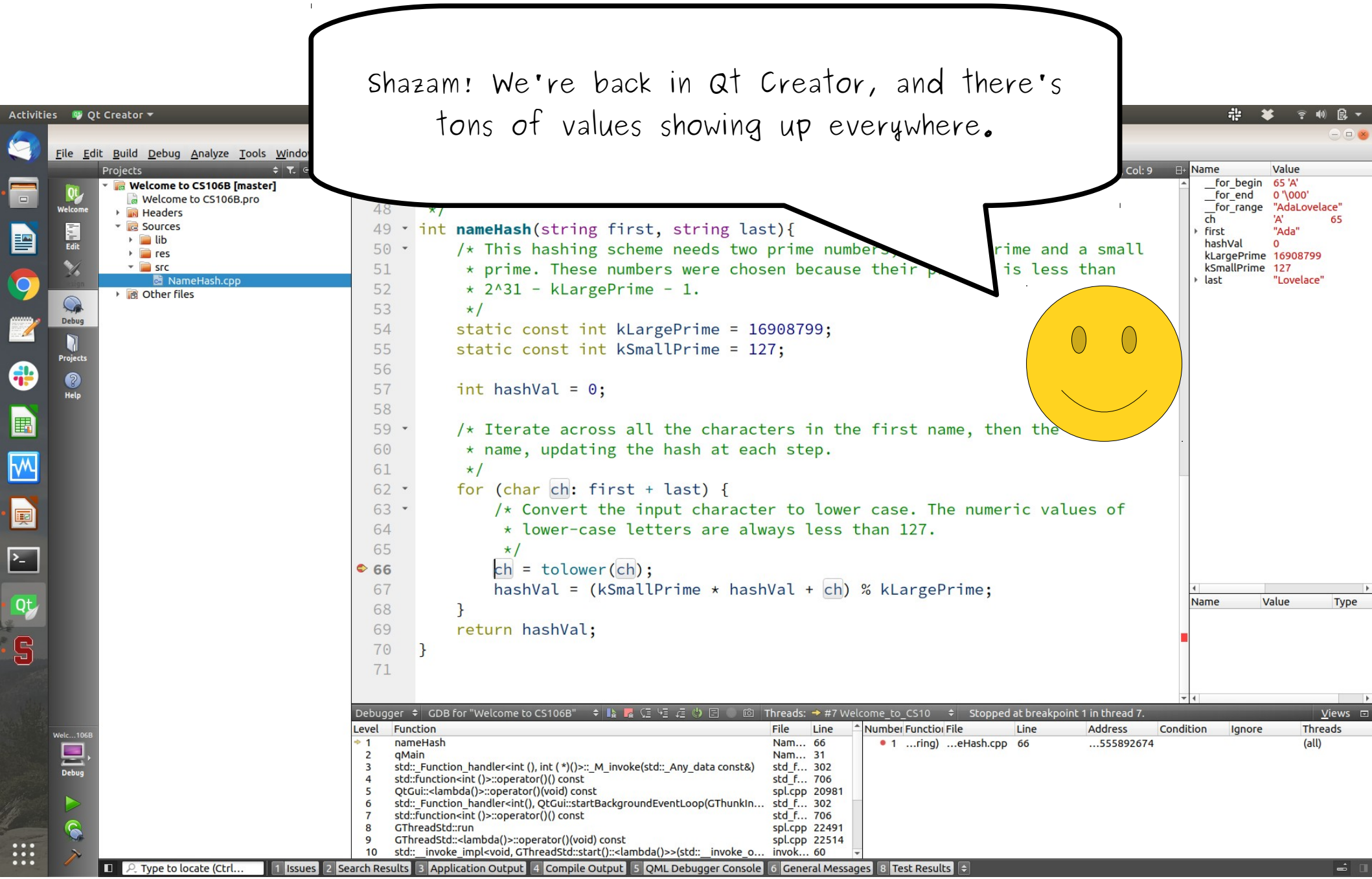
Now, type in **Lovelace** as a last name, but don't hit enter yet!

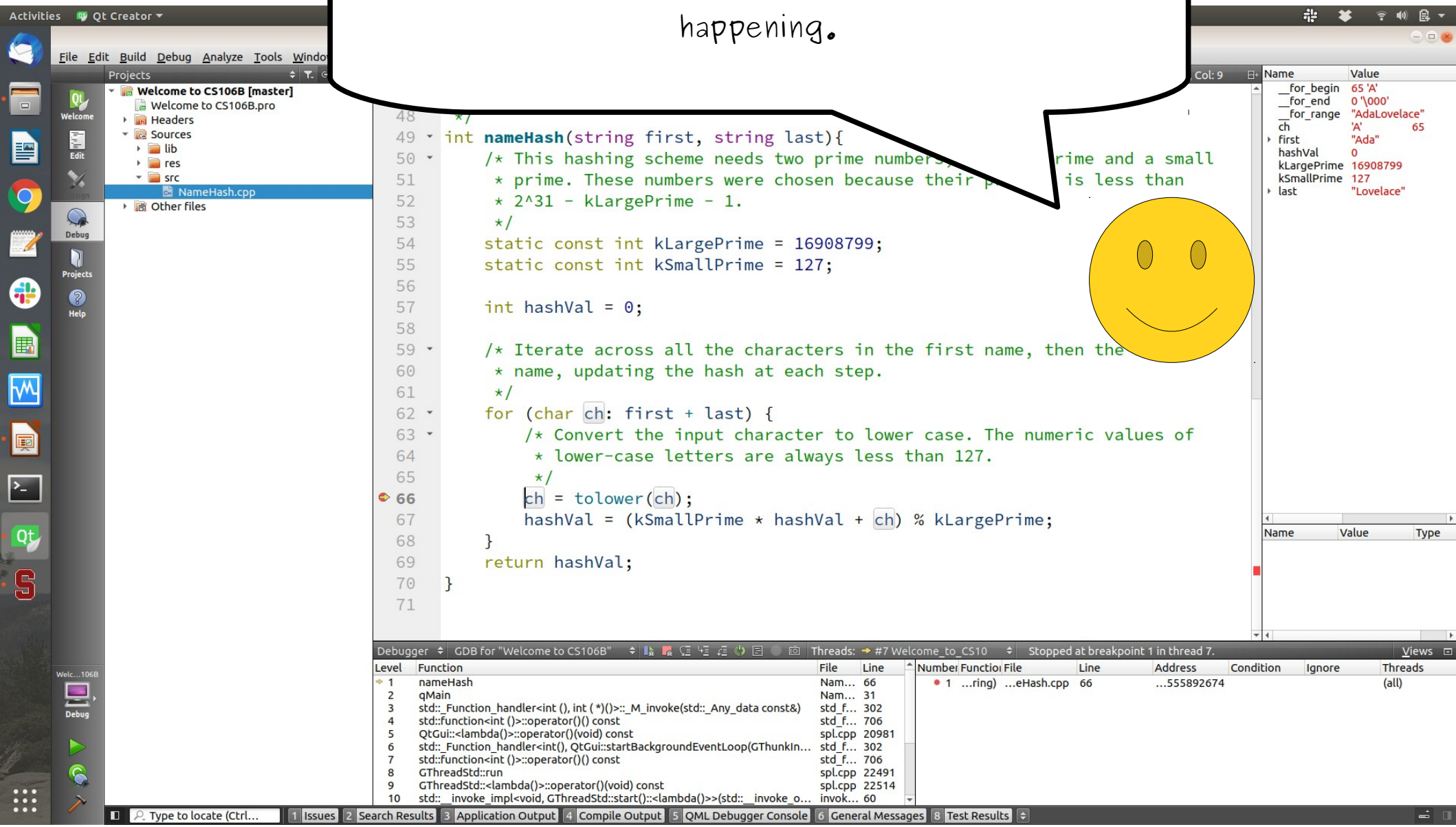


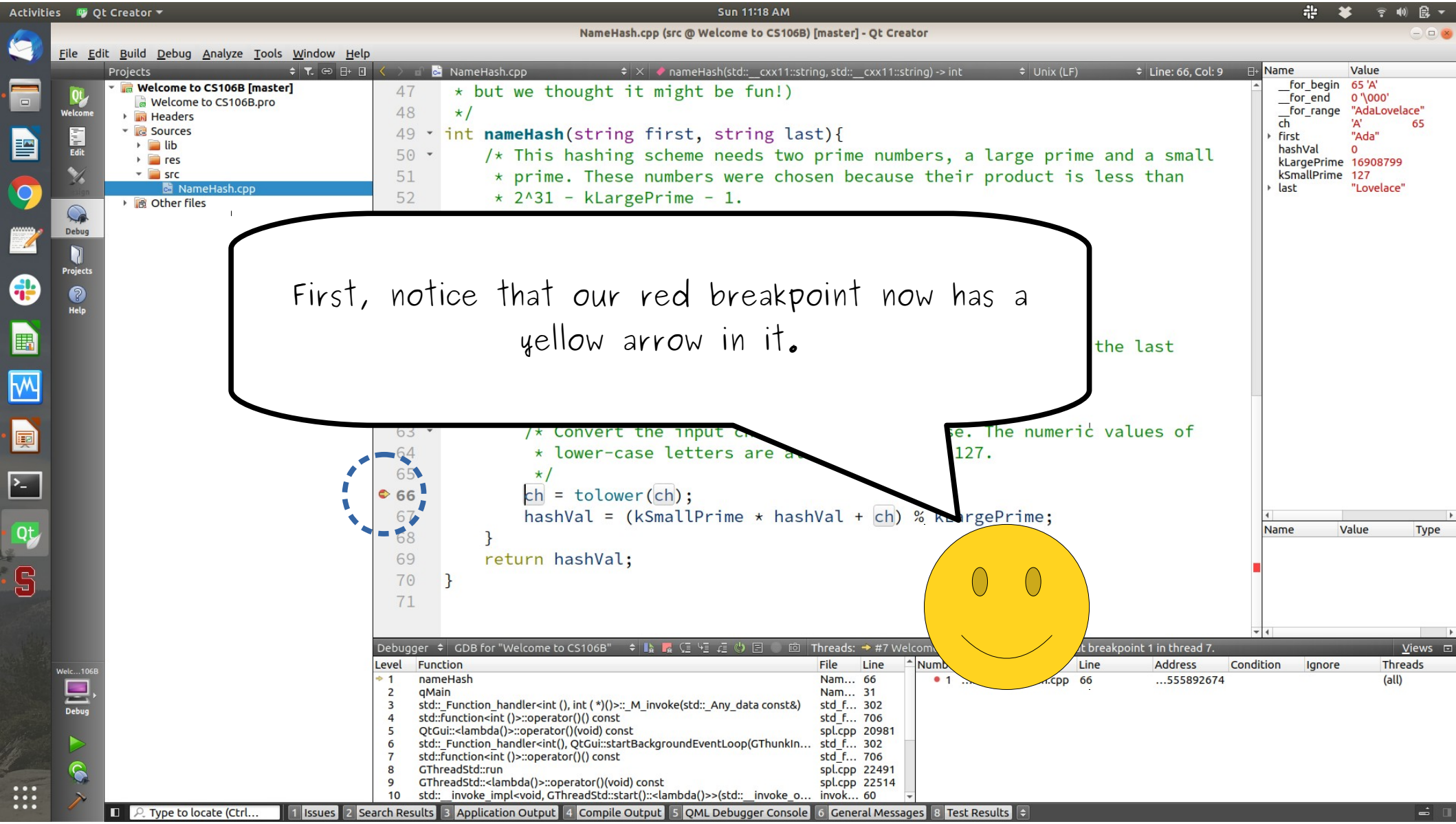
As soon as you hit enter, a bunch of things are going to pop up in Qt Creator. Don't panic! It's normal.

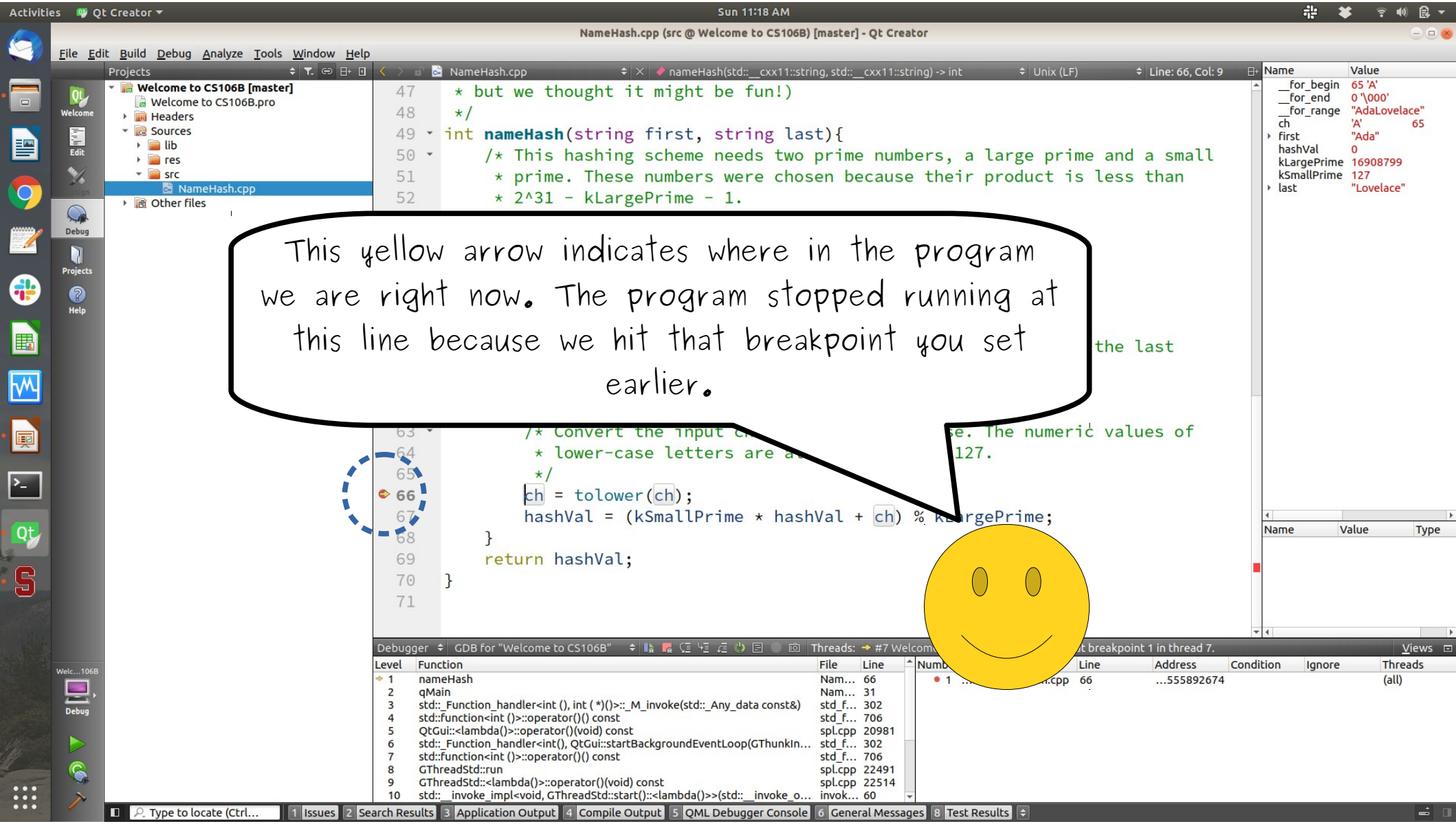


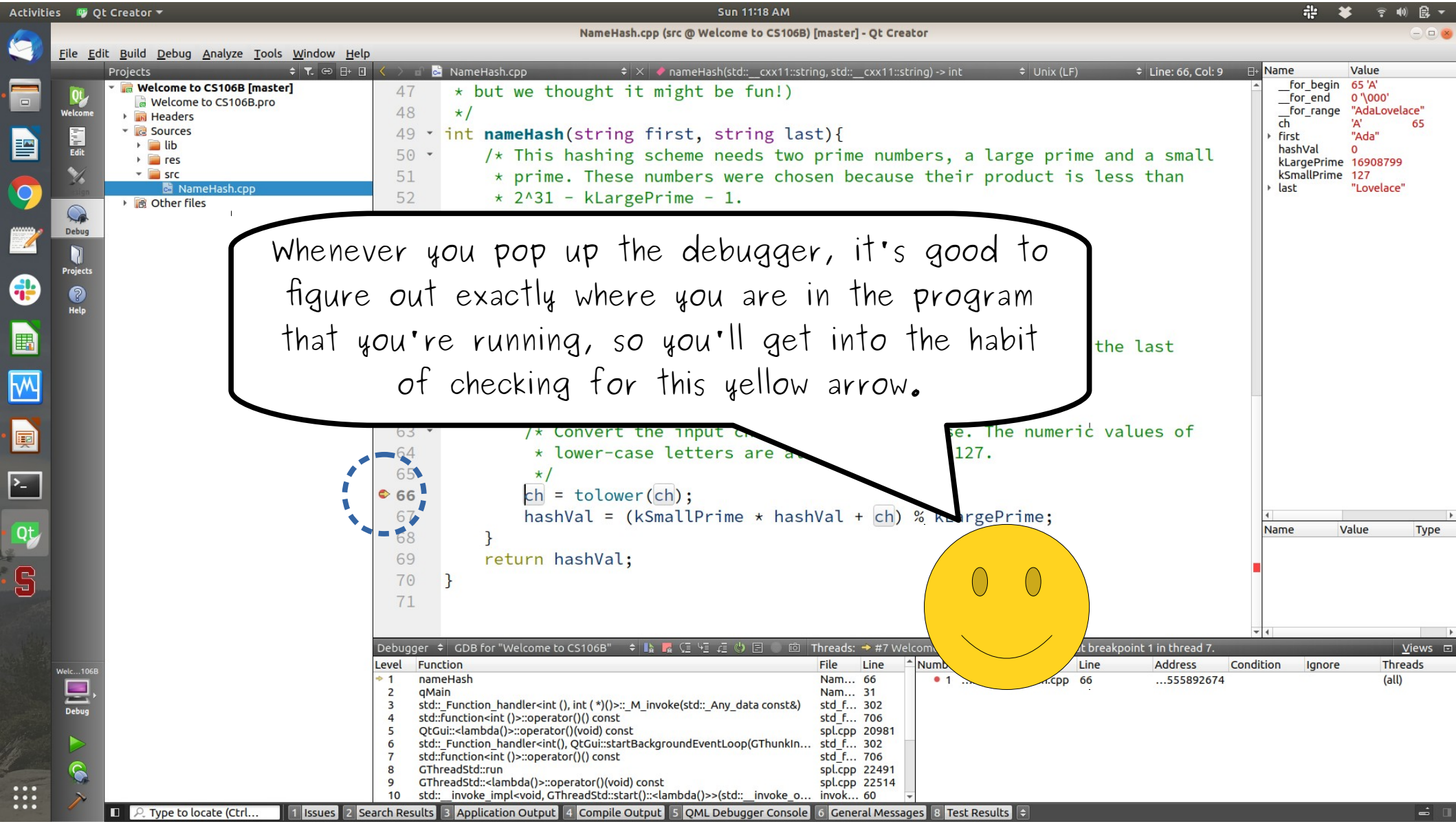

With that said, hit enter,
and watch the magic happen!
Activities
Welcome_to_CS106B
File Edit Build Debug Analyze Tools
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
Welcome
Edit
Debug
Projects
Help
Col: 43
Name Value Type
int nameHash(string first, string last){
/* This hashing scheme needs two prime numbers,
* prime. These numbers were chosen because their
Console
File Edit Options Help
What is your first name? Ada
What is your last name? Lovelace
Debugger
GDB for "Welcome to CS106B"
Application started.
Views
Level Function File Line
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

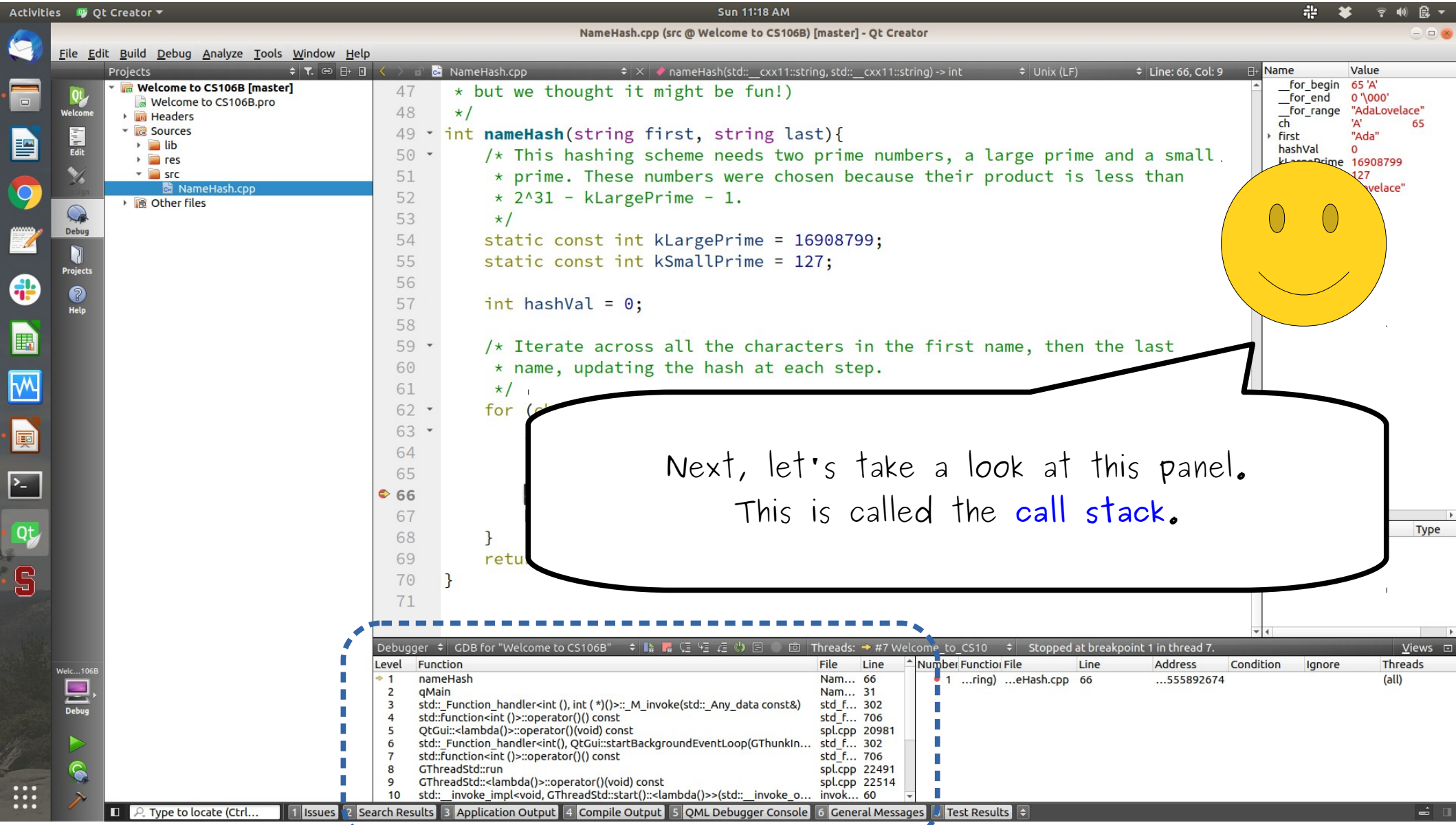

Shazam! We're back in Qt Creator, and there's tons of values showing up everywhere.
Activities
Qt Creator
File Edit Build Debug Analyze Tools
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
Welcome
Edit
Debug
Projects
Help
Col: 9
*/
int nameHash(string first, string last){
/* This hashing scheme needs two prime numbers, a large prime and a small
* prime. These numbers were chosen because their product is less than
* 2^31 - kLargePrime - 1.
*/
static const int kLargePrime = 16908799;
static const int kSmallPrime = 127;
int hashVal = 0;
/* Iterate across all the characters in the first name, then the
* name, updating the hash at each step.
*/
for (char ch: first + last) {
/* Convert the input character to lower case. The numeric values of
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

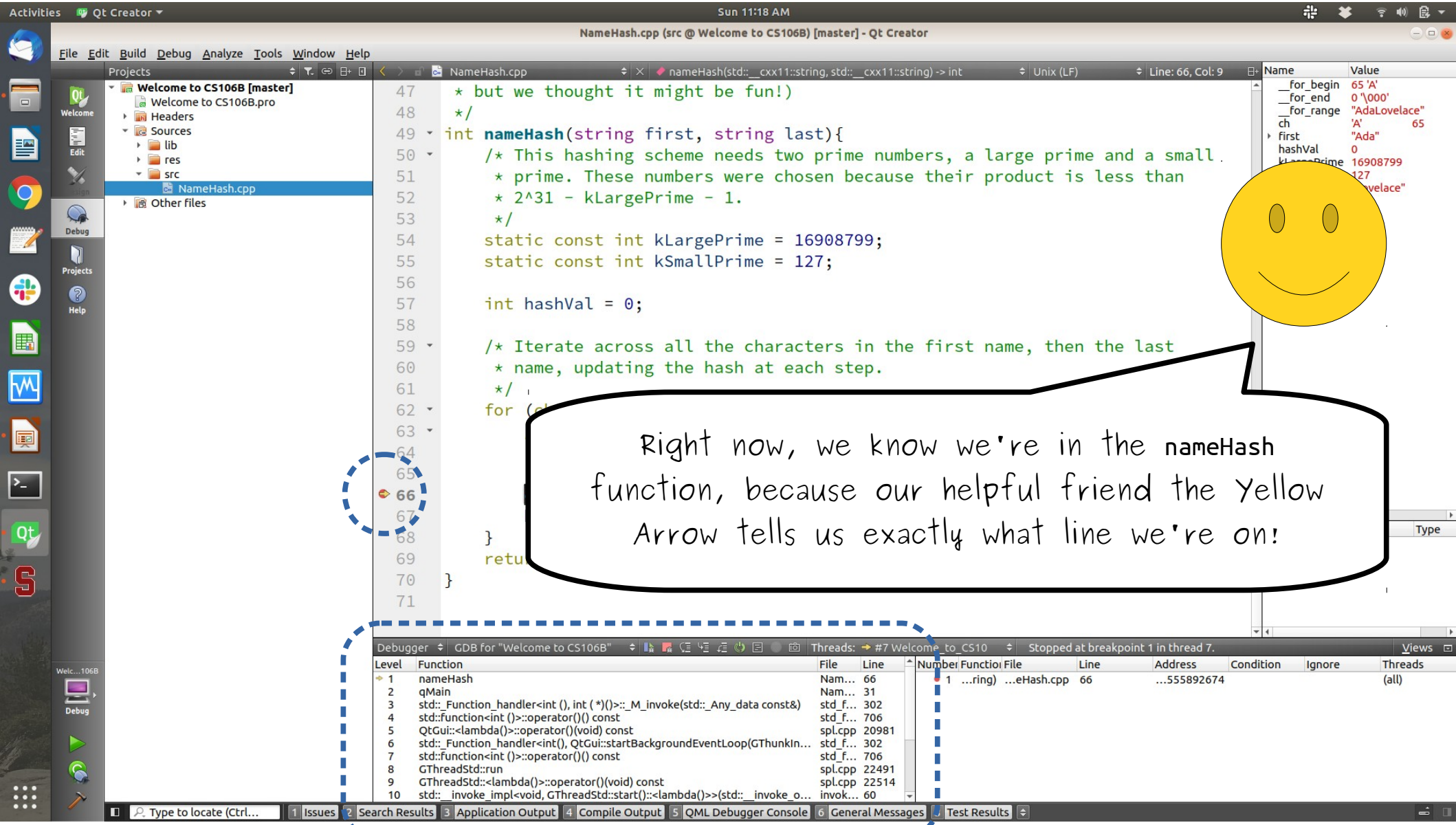

There's a lot going on right here. Let's see what's happening.
Activities
Qt Creator
File Edit Build Debug Analyze Tools Windo
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
Welcome
Edit
Debug
Projects
Help
Col: 9
48 */
49 int nameHash(string 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 p is less than
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Welc...106B
Debug
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

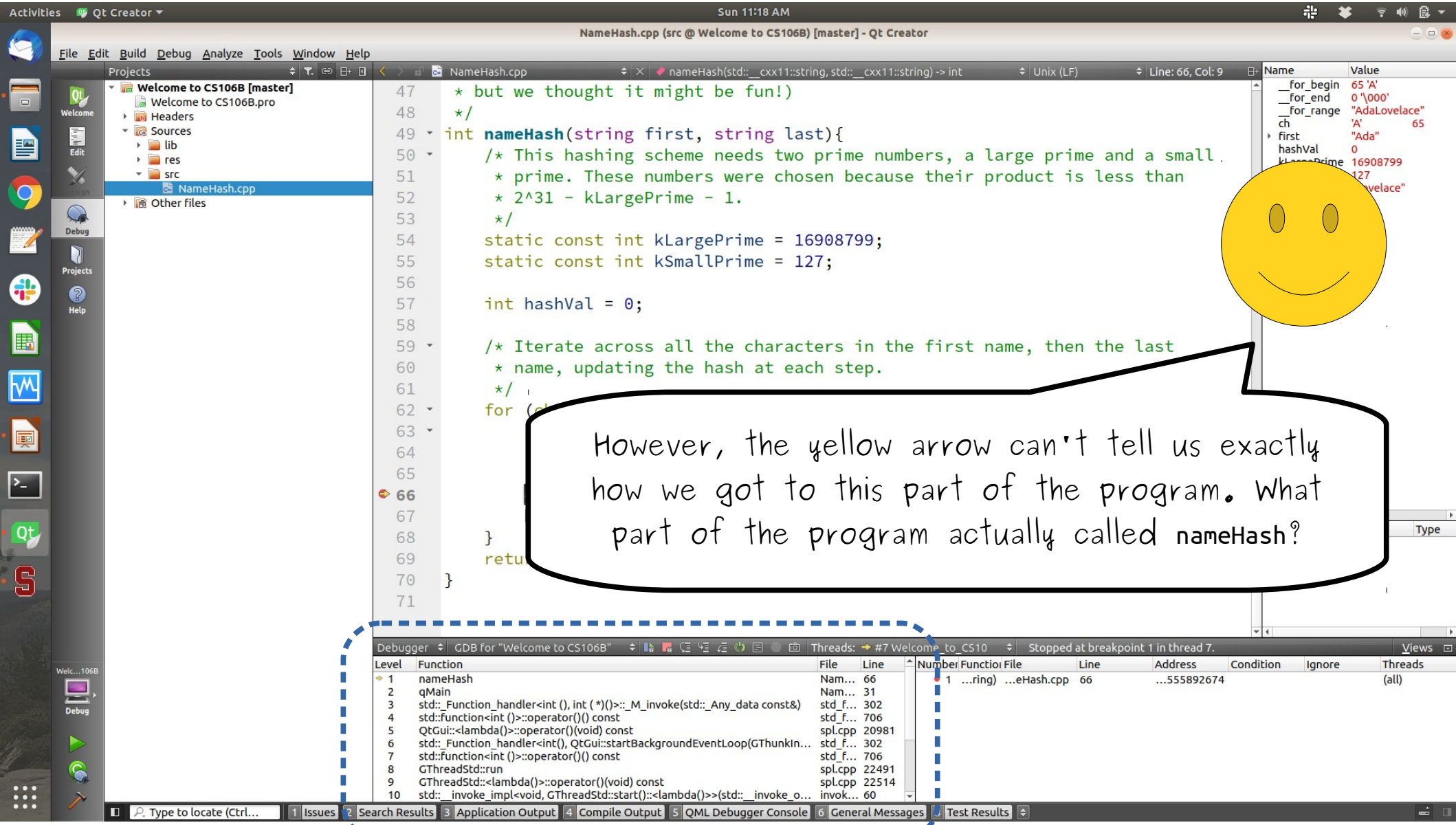

Activities
Qt Creator
Sun 11:18 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
63 /* Convert the input
64 * lower-case letters are
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) %
68 }
69 return hashVal;
70 }
71
the last
The numeric values of
127.
First, notice that our red breakpoint now has a yellow arrow in it.
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
GDB for "Welcome to CS106B"
Threads:
breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Line Address Condition Ignore Threads
66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

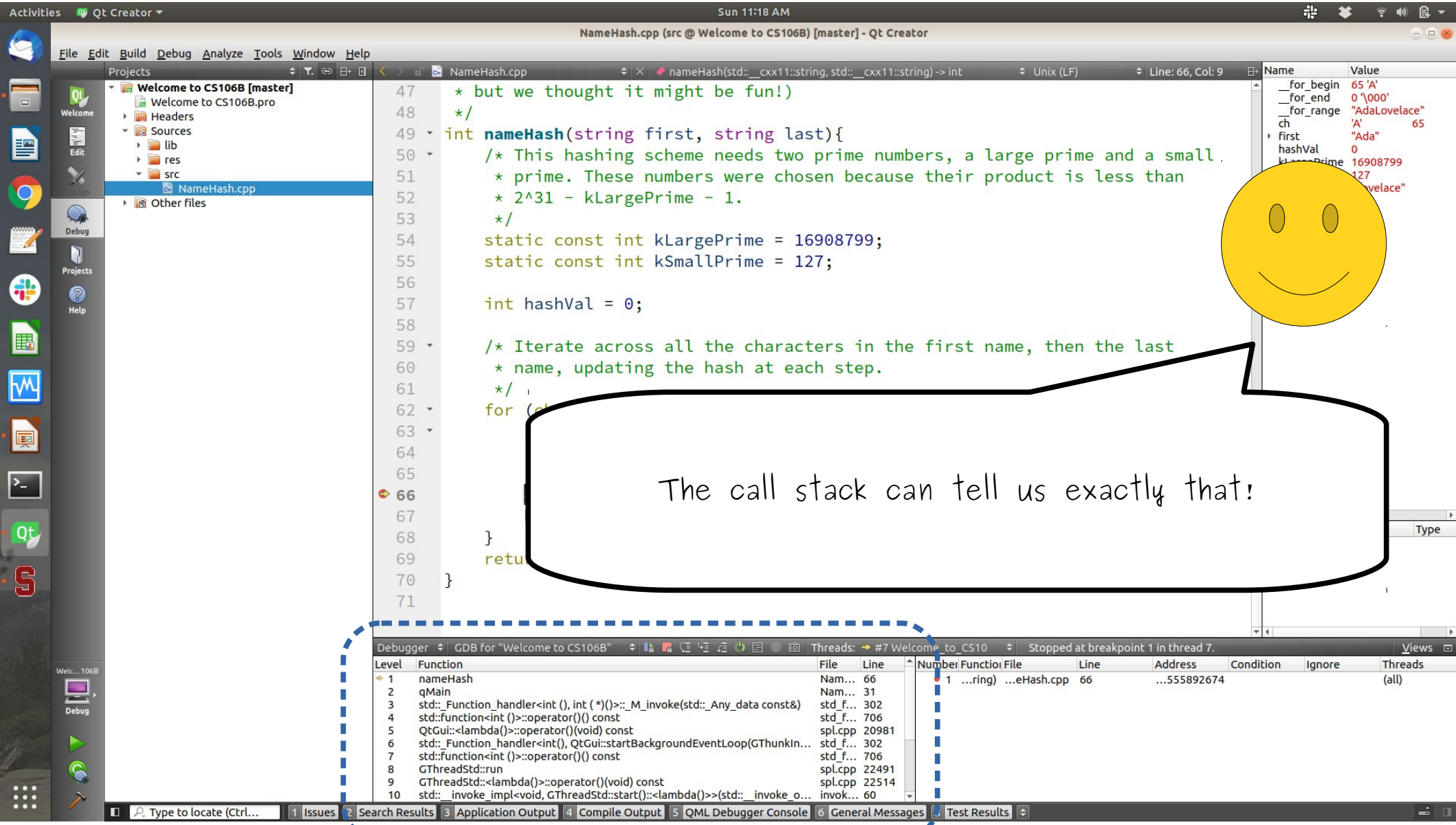

Activities
Qt Creator
Sun 11:18 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
the last
/* Convert the input
* lower-case letters are
127.
*/
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
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.
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
GDB for "Welcome to CS106B"
Threads:
breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Numb Line Address Condition Ignore Threads
1 ...cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

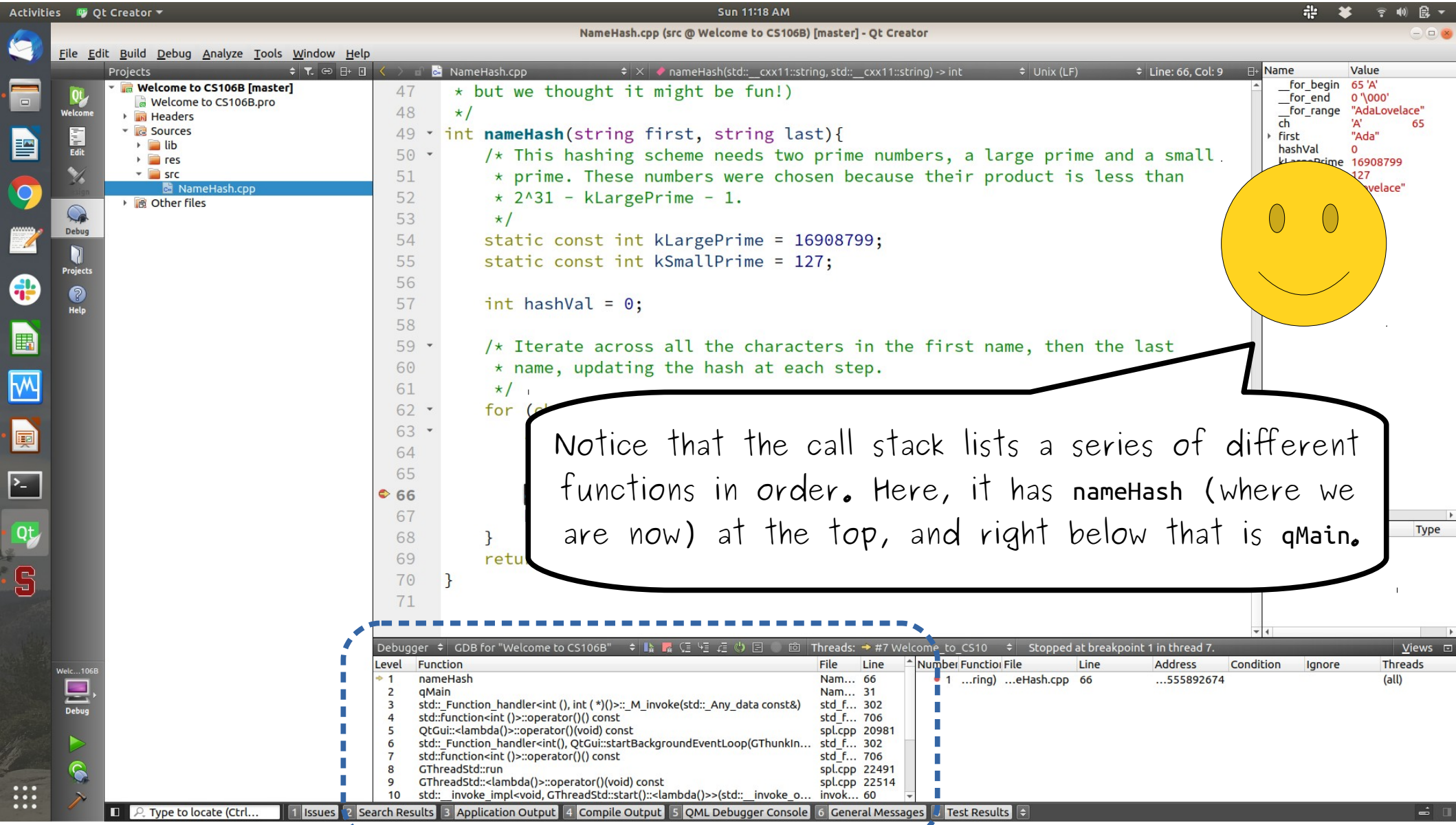

Activities
Qt Creator
Sun 11:18 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
63 /* Convert the input
64 * lower-case letters are
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
the last
e. The numeric values of
127.
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.
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcom
breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Numb
Line Address Condition Ignore Threads
1 ...cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results
Welcome
Edit
Design
Debug
Projects
Help
Welc...106B
Debug

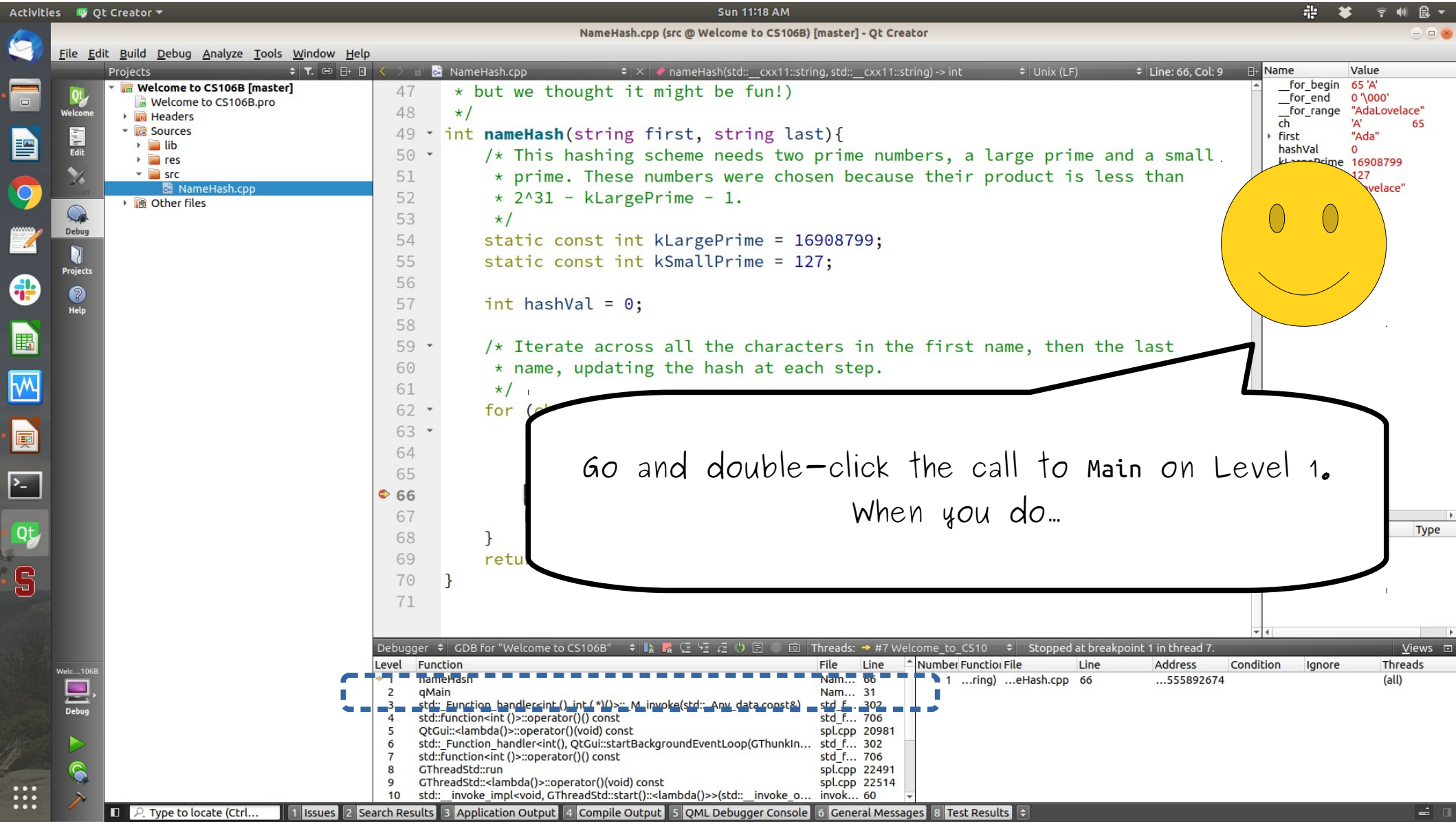

Activities
Qt Creator
Sun 11:18 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
127
Next, let's take a look at this panel.
This is called the call stack.
Debugger GDB for "Welcome to CS106B" Threads: #7 Welcome_to_CS10 Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages Test Results

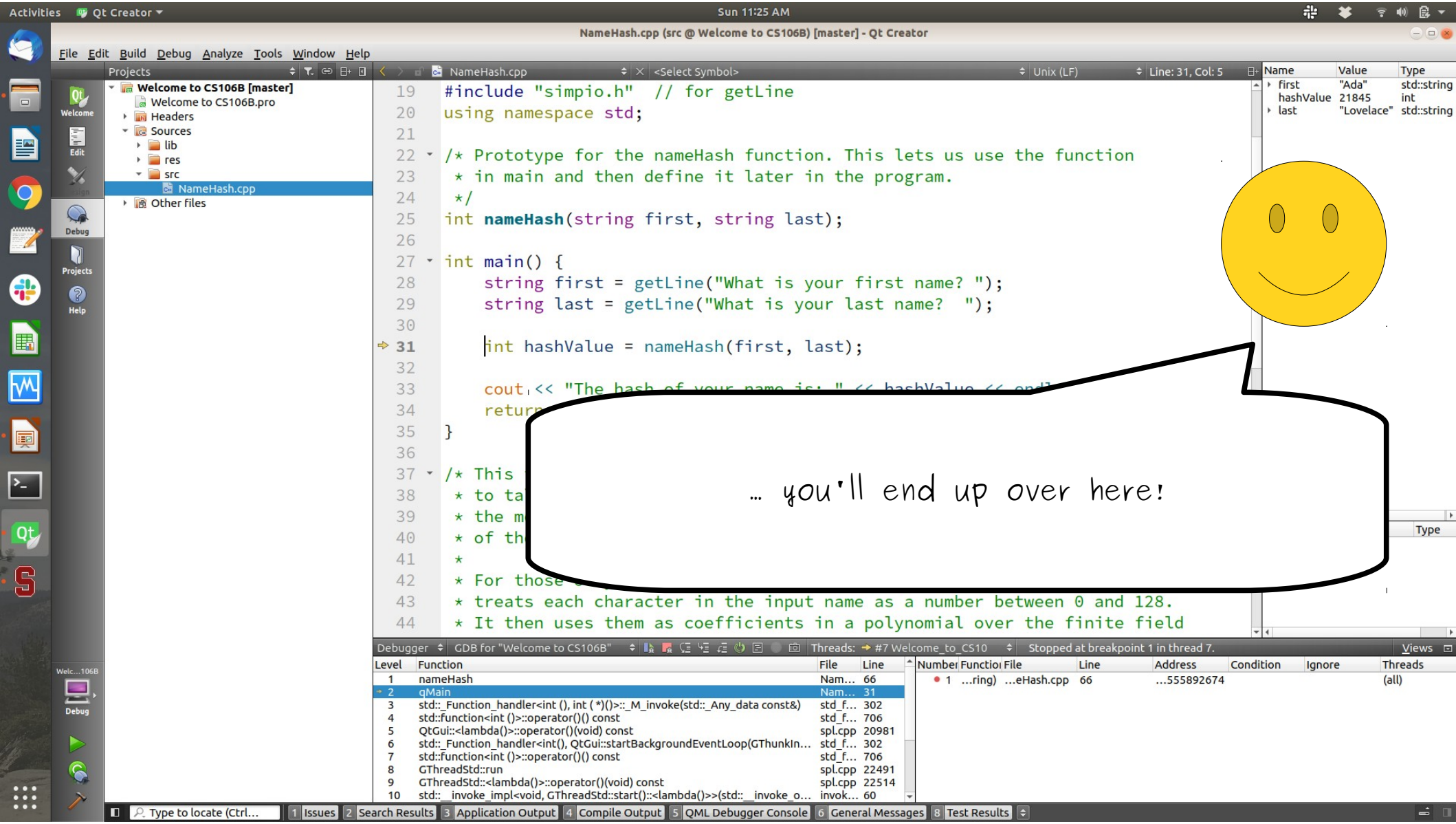

Activities
Qt Creator
Sun 11:18 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
16908799
127
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!
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
Test Results

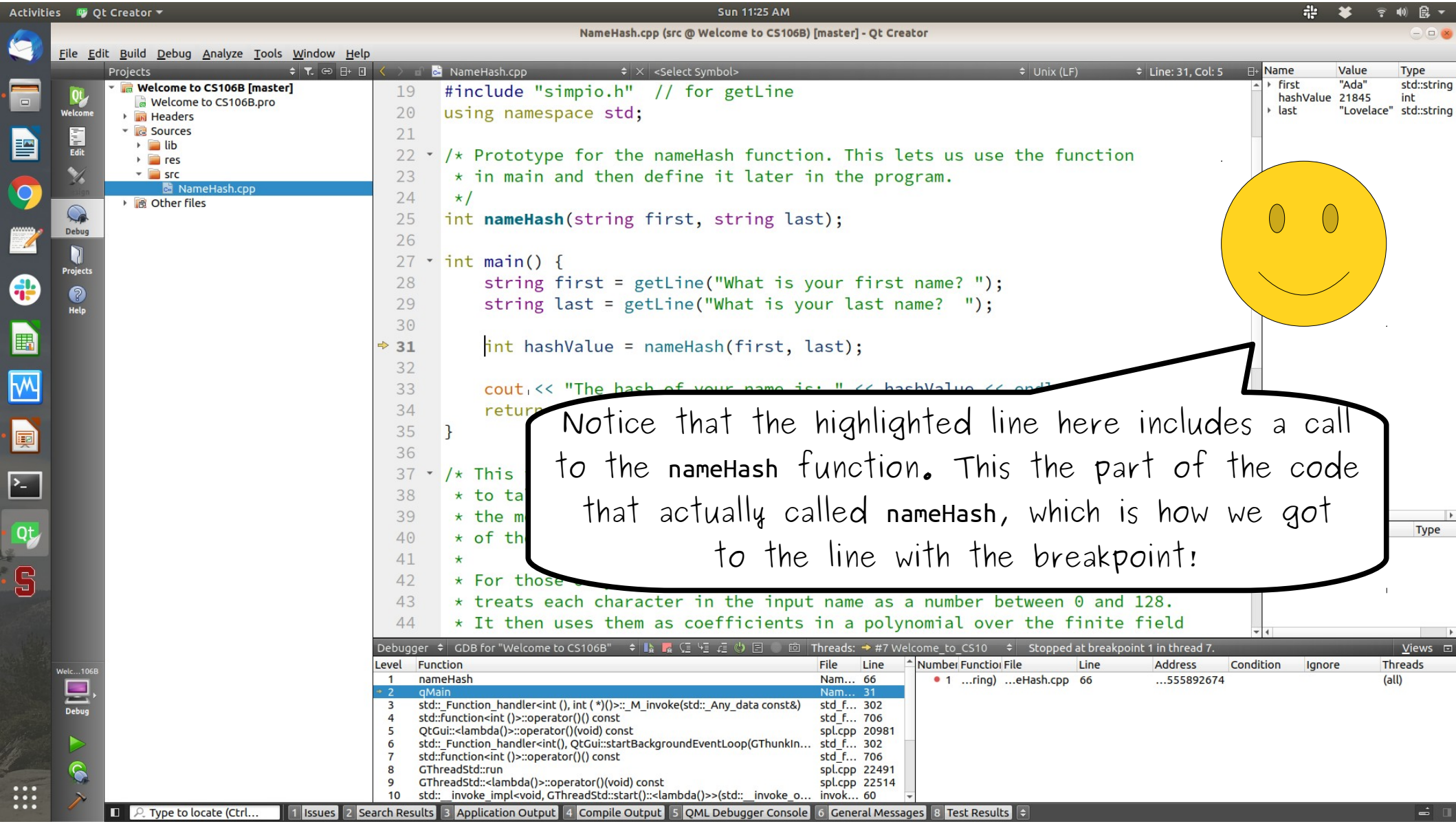

Activities
Qt Creator
Sun 11:18 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name
Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
16908799
127
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?
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
Test Results

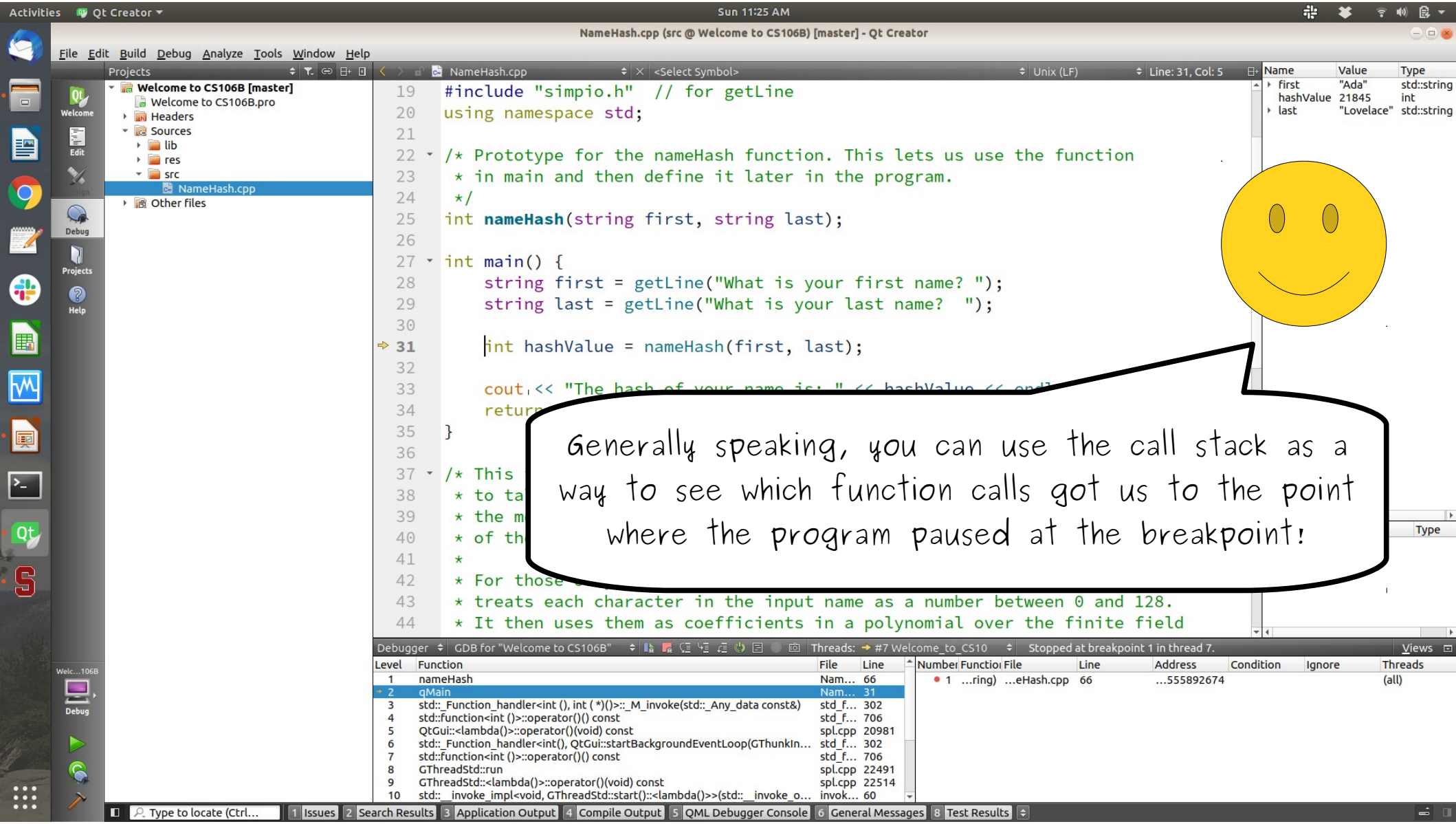

Activities
Qt Creator
Sun 11:18 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
16908799
127
Type
The call stack can tell us exactly that!
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
Test Results

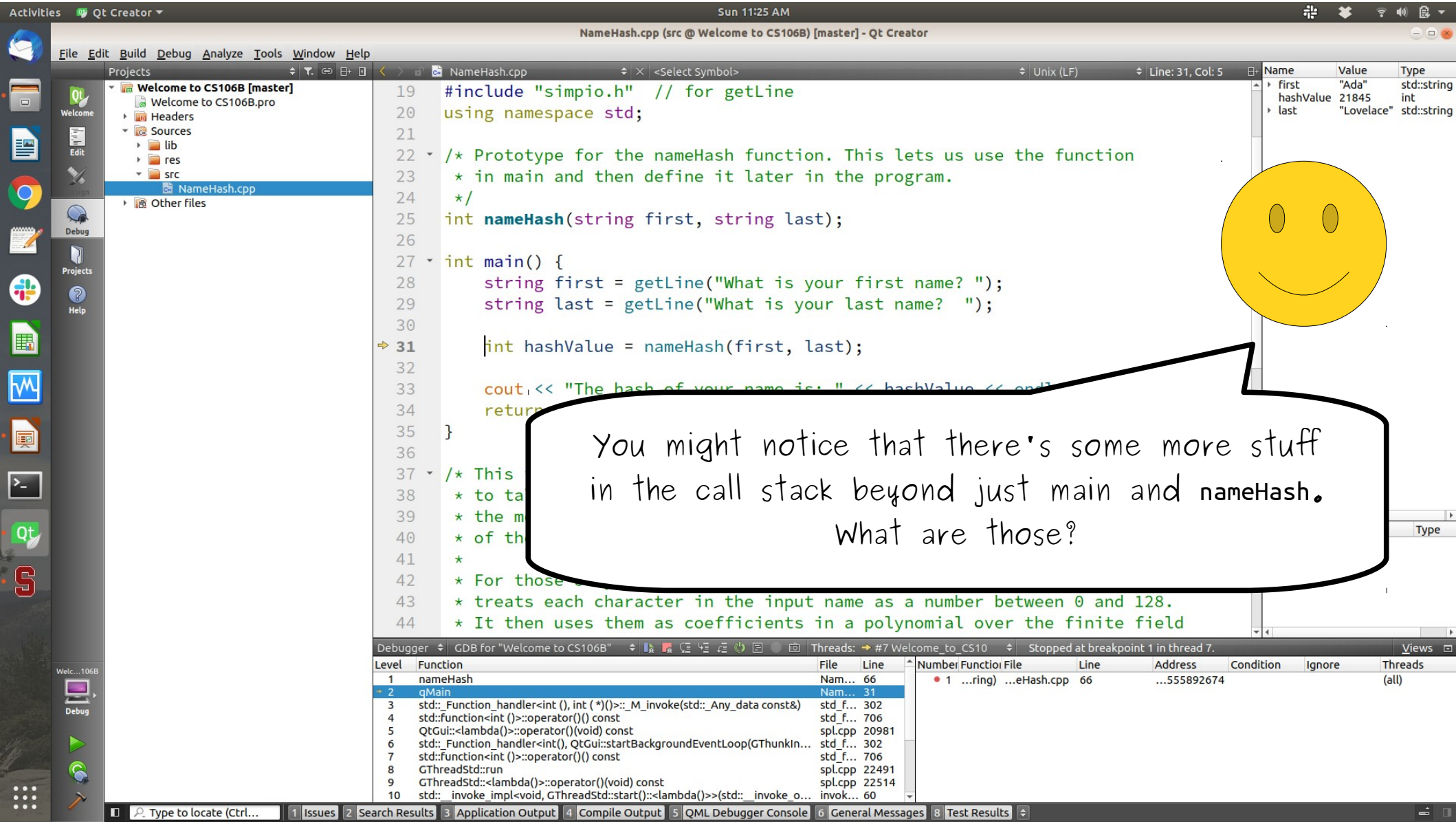

Activities
Qt Creator
Sun 11:18 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
16908799
127
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 qMain.
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
Test Results

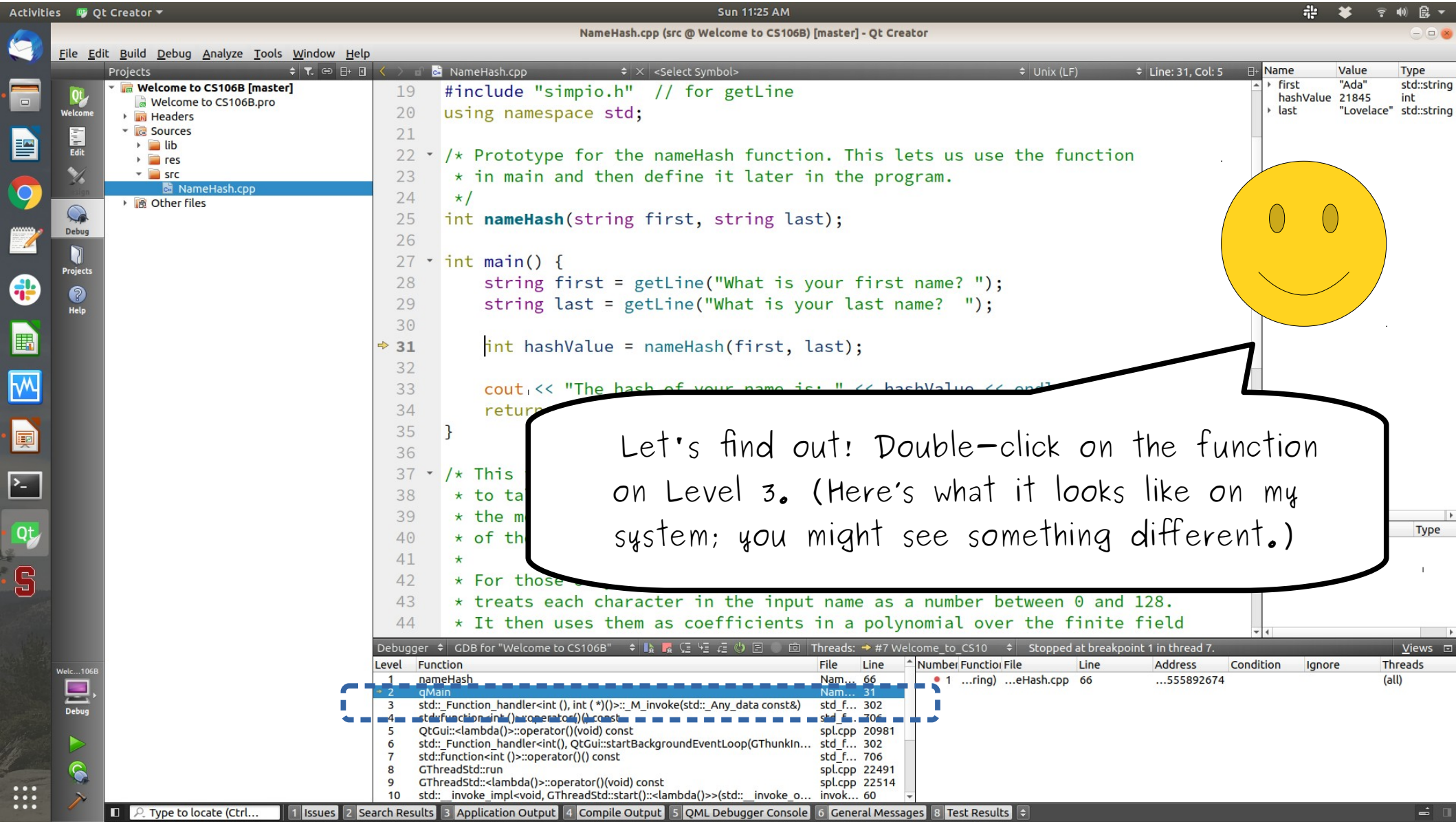

Activities
Qt Creator
Sun 11:18 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name
Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
16908799
127
Type
Go and double-click the call to Main on Level 1.
When you do...
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

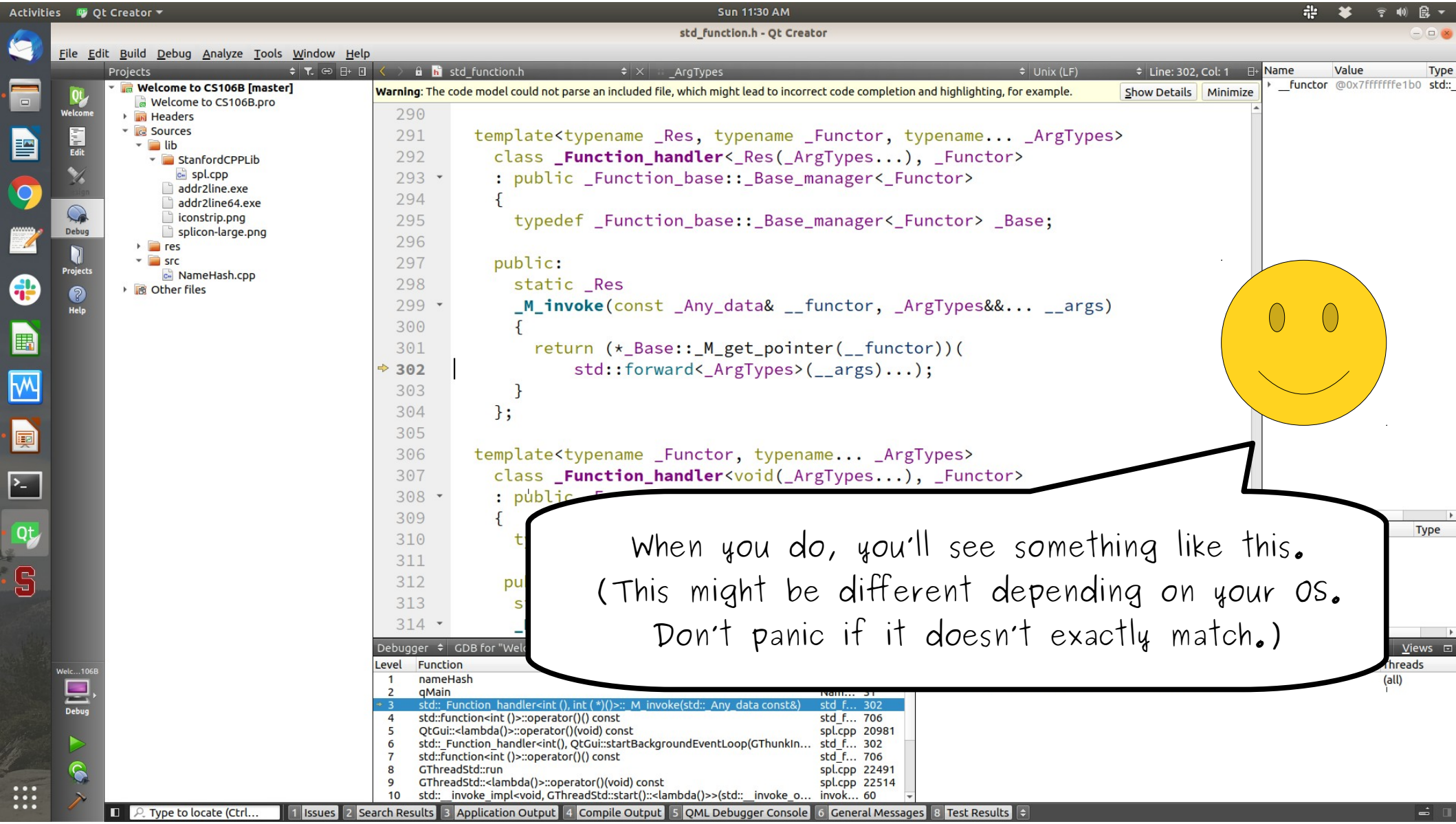

Activities
Qt Creator
Sun 11:25 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 5
#include "simpio.h" // for getLine
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);
cout << "The hash of your name is: " << hashValue << endl;
return
}
/* This
* to ta
* the m
* of th
*
* For those
* treats each character in the input name as a number between 0 and 128.
* It then uses them as coefficients in a polynomial over the finite field
Name Value Type
first "Ada" std::string
hashValue 21845 int
last "Lovelace" std::string
… you'll end up over here!
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam… 66
2 qMain Nam… 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f… 302
4 std::function<int ()>::operator()() const std_f… 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn… std_f… 302
7 std::function<int ()>::operator()() const std_f… 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o… invok… 60
Number Functio File Line Address Condition Ignore Threads
1 …ring) …eHash.cpp 66 …555892674 (all)
Type
Type to locate (Ctrl…
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

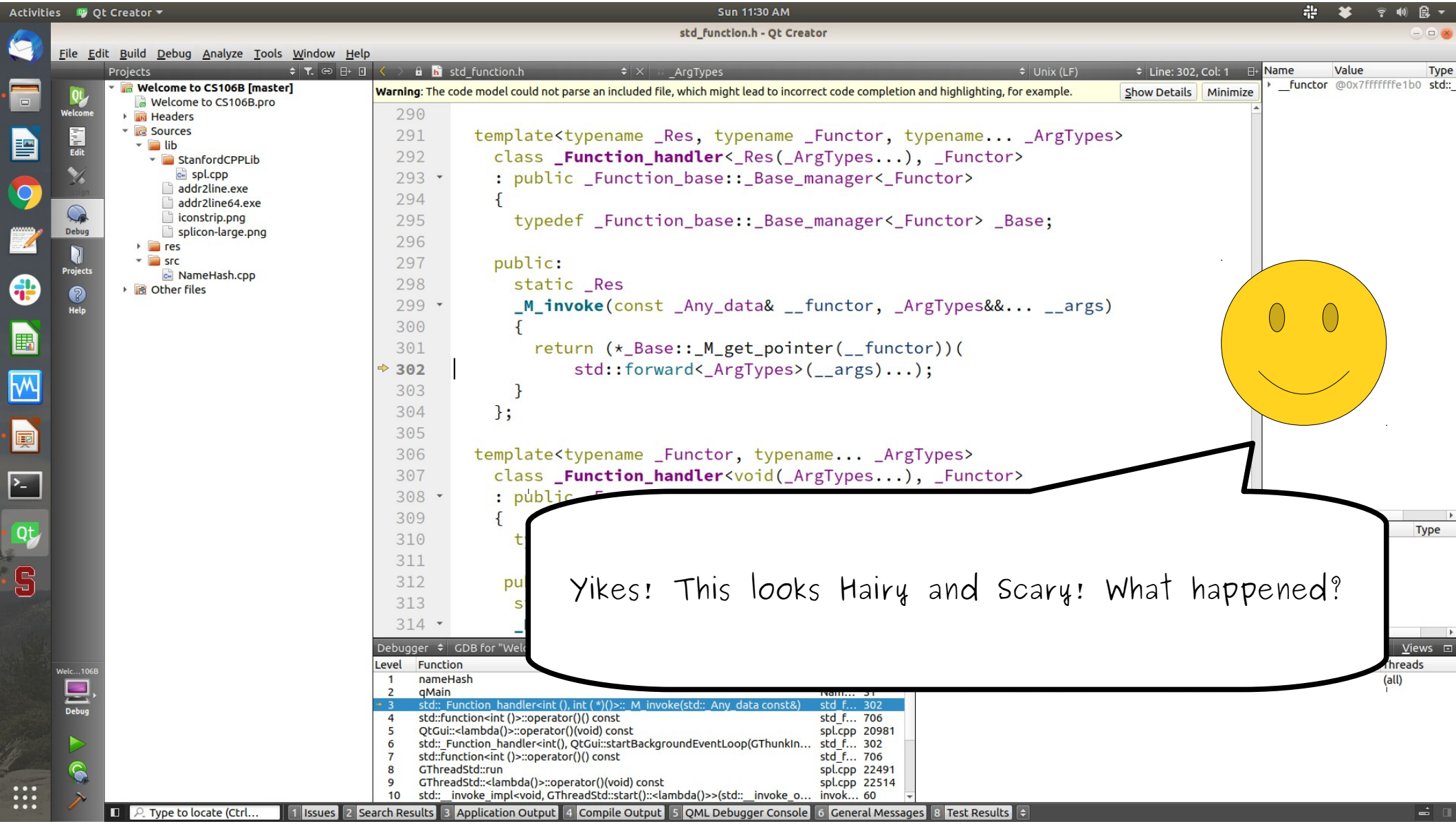

Activities
Qt Creator
Sun 11:25 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 5
19 #include "simpio.h" // for getLine
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);
32
33 cout << "The hash of your name is: " << hashValue << endl;
34 return 0;
35 }
36
37 /* This
38 * to ta
39 * the m
40 * of th
41 *
42 * For those
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
Name Value Type
first "Ada" std::string
hashValue 21845 int
last "Lovelace" std::string
Notice that the highlighted line here includes a call to the nameHash function. This the part of the code that actually called nameHash, which is how we got to the line with the breakpoint!
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

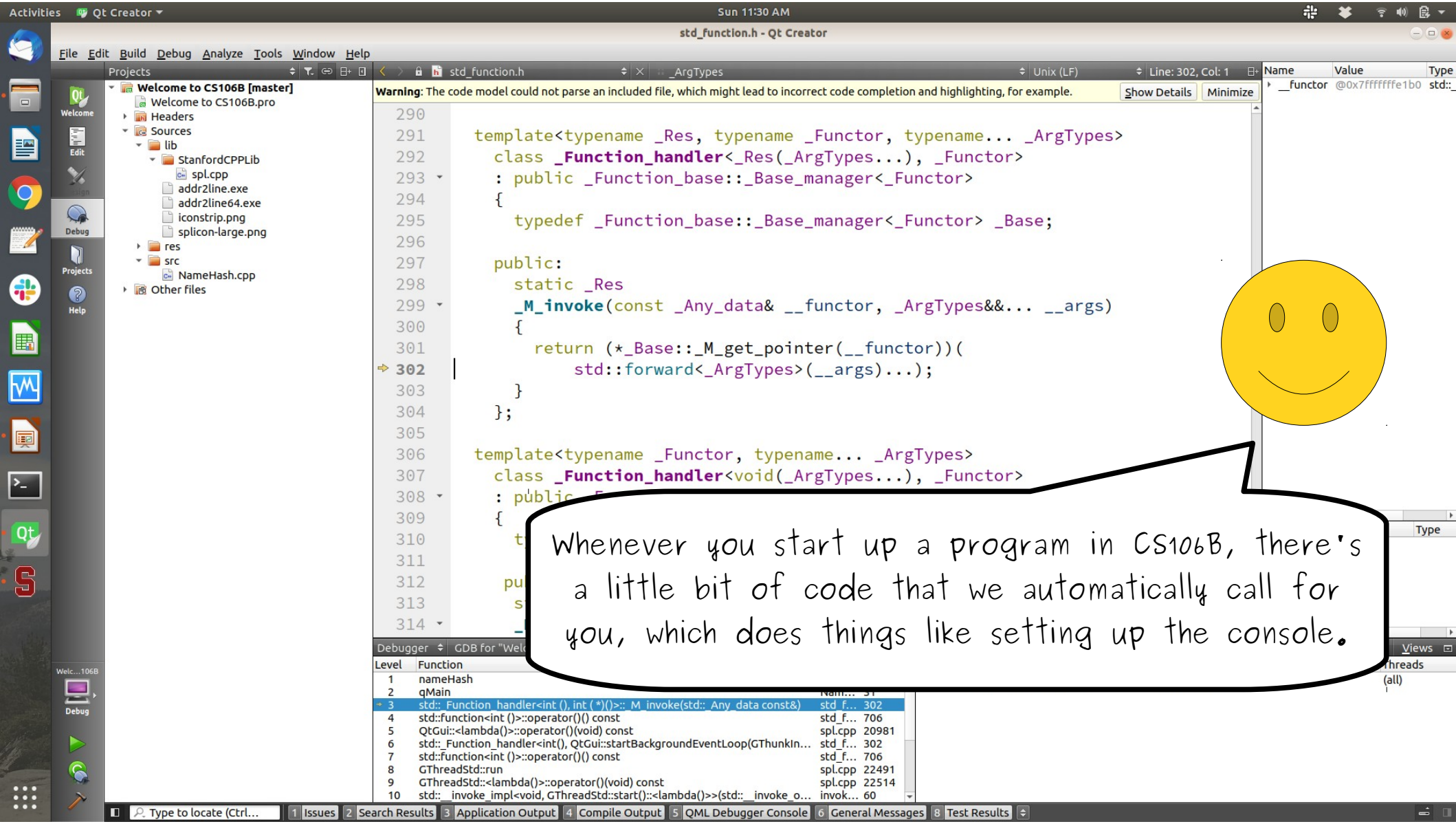

Activities
Qt Creator
Sun 11:25 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 5
#include "simpio.h" // for getLine
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);
* treats each character in the input name as a number between 0 and 128.
* It then uses them as coefficients in a polynomial over the finite field
Name Value Type
first "Ada" std::string
hashValue 21845 int
last "Lovelace" std::string
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!
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

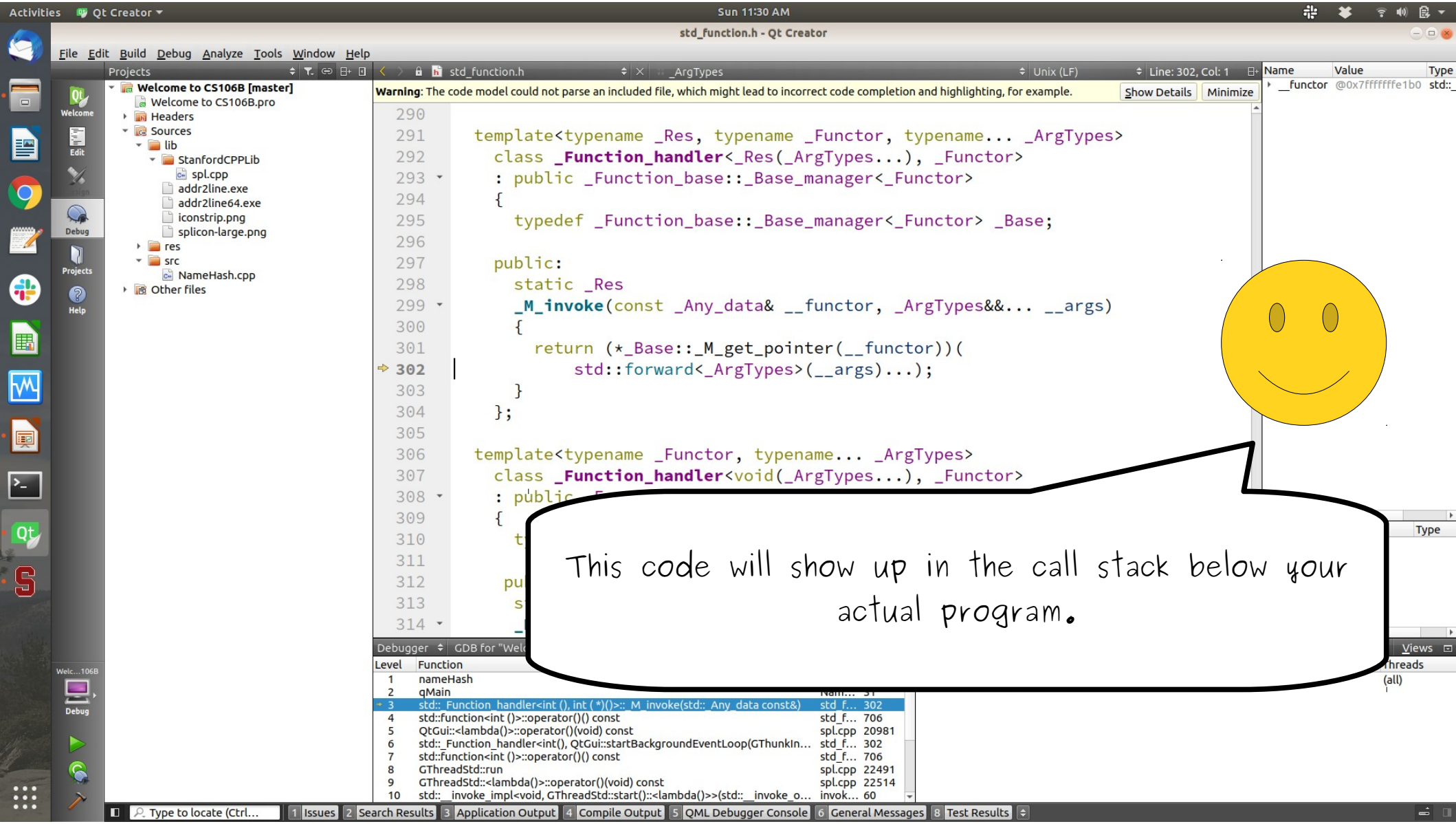

Activities
Qt Creator
Sun 11:25 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 5
#include "simpio.h" // for getLine
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);
* treats each character in the input name as a number between 0 and 128.
* It then uses them as coefficients in a polynomial over the finite field
Name Value Type
first "Ada" std::string
hashValue 21845 int
last "Lovelace" std::string
You might notice that there's some more stuff in the call stack beyond just main and nameHash. What are those?
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

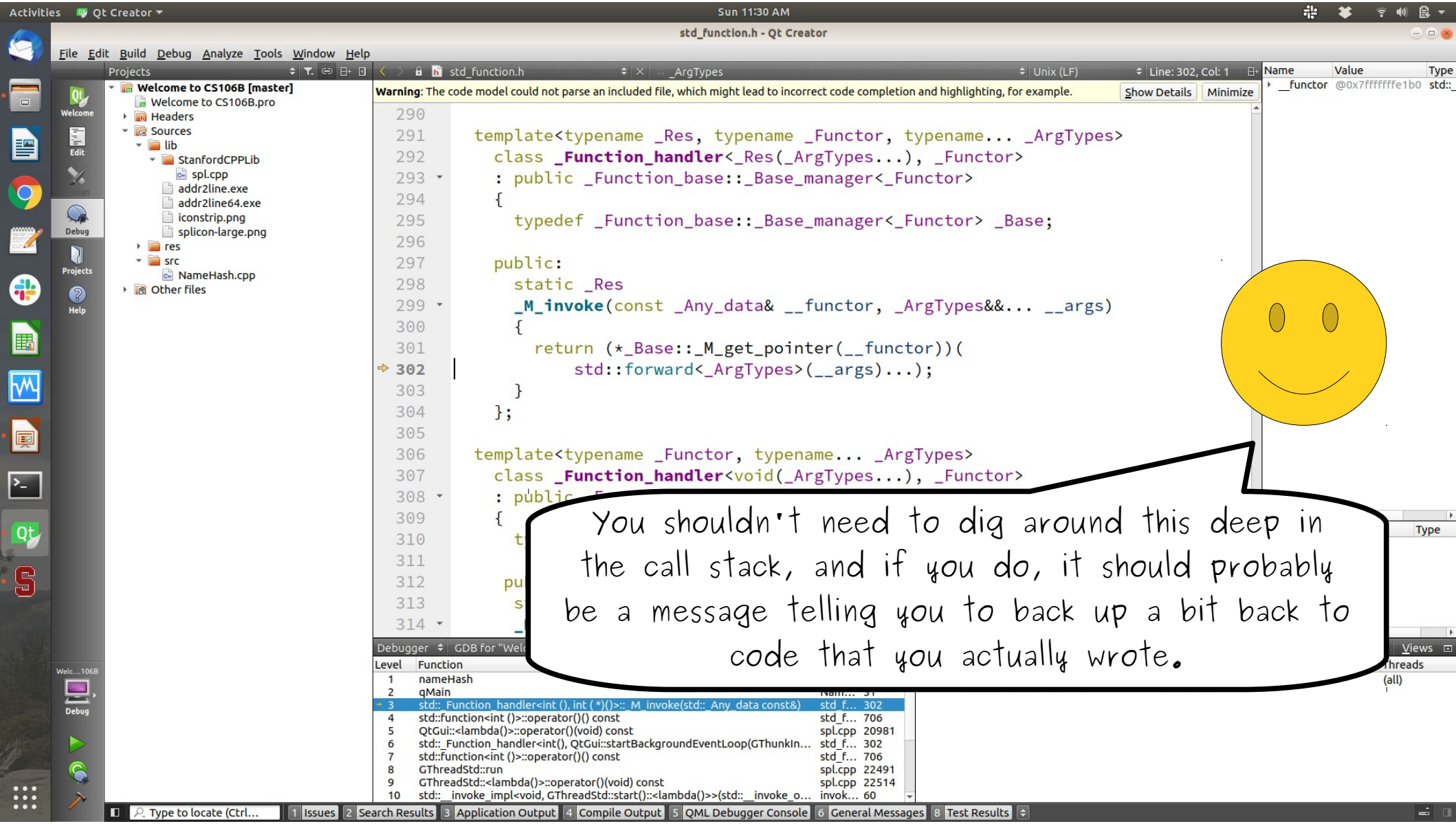

Activities
Qt Creator
Sun 11:25 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 5
#include "simpio.h" // for getLine
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);
* treats each character in the input name as a number between 0 and 128.
* It then uses them as coefficients in a polynomial over the finite field
Name Value Type
first "Ada" std::string
hashValue 21845 int
last "Lovelace" std::string
Let's find out! Double-click on the function on Level 3. (Here's what it looks like on my system; you might see something different.)
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results
Welc...106B
Debug


Activities
Qt Creator
Sun 11:30 AM
std_function.h - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
std_function.h
_ArgTypes
Unix (LF)
Line: 302, Col: 1
Warning: The code model could not parse an included file, which might lead to incorrect code completion and highlighting, for example.
Show Details
Minimize
Name
Value
Type
__functor
@0x7fffffffe1b0
std::_
template<typename _Res, typename _Functor, typename... _ArgTypes>
class _Function_handler<_Res(_ArgTypes...), _Functor>
: public _Function_base::_Base_manager<_Functor>
{
typedef _Function_base::_Base_manager<_Functor> _Base;
public:
static _Res
_M_invoke(const _Any_data& __functor, _ArgTypes&&... __args)
{
return (*_Base::_M_get_pointer(__functor))(
std::forward<_ArgTypes>(__args)...);
}
};
template<typename _Functor, typename... _ArgTypes>
class _Function_handler<void(_ArgTypes...), _Functor>
: public
{
When you do, you'll see something like this.
(This might be different depending on your OS.
Don't panic if it doesn't exactly match.)
Debugger
GDB for "Wel
Views
Level
Function
Type
Threads
(all)
1 nameHash
2 qMain
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results


Activities
Qt Creator
Sun 11:30 AM
std_function.h - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
std_function.h
_ArgTypes
Unix (LF)
Line: 302, Col: 1
Warning: The code model could not parse an included file, which might lead to incorrect code completion and highlighting, for example.
Show Details
Minimize
Name
Value
Type
__functor
@0x7fffffffe1b0
std::_
template<typename _Res, typename _Functor, typename... _ArgTypes>
class _Function_handler<_Res(_ArgTypes...), _Functor>
: public _Function_base::_Base_manager<_Functor>
{
typedef _Function_base::_Base_manager<_Functor> _Base;
public:
static _Res
_M_invoke(const _Any_data& __functor, _ArgTypes&&... __args)
{
return (*_Base::_M_get_pointer(__functor))(
std::forward<_ArgTypes>(__args)...);
}
};
template<typename _Functor, typename... _ArgTypes>
class _Function_handler<void(_ArgTypes...), _Functor>
: public
{
Yikes! This looks Hairy and Scary! What happened?
Debugger
GDB for "Wel
Views
Level
Function
Type
Threads
(all)
1 nameHash
2 qMain
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results


Activities
Qt Creator
Sun 11:30 AM
std_function.h - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
std_function.h
_ArgTypes
Unix (LF)
Line: 302, Col: 1
Warning: The code model could not parse an included file, which might lead to incorrect code completion and highlighting, for example.
Show Details
Minimize
template<typename _Res, typename _Functor, typename... _ArgTypes>
class _Function_handler<_Res(_ArgTypes...), _Functor>
: public _Function_base::_Base_manager<_Functor>
{
typedef _Function_base::_Base_manager<_Functor> _Base;
public:
static _Res
_M_invoke(const _Any_data& __functor, _ArgTypes&&... __args)
{
return (*_Base::_M_get_pointer(__functor))(
std::forward<_ArgTypes>(__args)...);
}
};
template<typename _Functor, typename... _ArgTypes>
class _Function_handler<void(_ArgTypes...), _Functor>
: public
{
Name
Value
Type
__functor
@0x7fffffffe1b0
std::
Whenever you start up a program in CS106B, there's a little bit of code that we automatically call for you, which does things like setting up the console.
Debugger
GDB for "Wel
Level
Function
1 nameHash
2 qMain
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Views
Threads
(all)
Welc...106B
Debug
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results


Activities
Qt Creator
Sun 11:30 AM
std_function.h - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
std_function.h
_ArgTypes
Unix (LF)
Line: 302, Col: 1
Warning: The code model could not parse an included file, which might lead to incorrect code completion and highlighting, for example.
Show Details
Minimize
Name
Value
Type
__functor
@0x7fffffffe1b0
std::_
290
291 template<typename _Res, typename _Functor, typename... _ArgTypes>
292 class _Function_handler<_Res(_ArgTypes...), _Functor>
293 : public _Function_base::_Base_manager<_Functor>
294 {
295 typedef _Function_base::_Base_manager<_Functor> _Base;
296
297 public:
298 static _Res
299 _M_invoke(const _Any_data& __functor, _ArgTypes&&... __args)
300 {
301 return (*_Base::_M_get_pointer(__functor))(
302 std::forward<_ArgTypes>(__args)...);
303 }
304 };
305
306 template<typename _Functor, typename... _ArgTypes>
307 class _Function_handler<void(_ArgTypes...), _Functor>
308 : public
309 {
310 t
311
312 pu
313 s
314
This code will show up in the call stack below your actual program.
Debugger
GDB for "Welc
Views
Level
Function
Type
Threads
(all)
1 nameHash
2 qMain
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results
Welcome
Edit
Debug
Projects
Help
Welc...106B
Debug


Activities
Qt Creator
Sun 11:30 AM
std_function.h - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
std_function.h
_ArgTypes
Unix (LF)
Line: 302, Col: 1
Warning: The code model could not parse an included file, which might lead to incorrect code completion and highlighting, for example.
Show Details
Minimize
290
291 template<typename _Res, typename _Functor, typename... _ArgTypes>
292 class _Function_handler<_Res(_ArgTypes...), _Functor>
293 : public _Function_base::_Base_manager<_Functor>
294 {
295 typedef _Function_base::_Base_manager<_Functor> _Base;
296
297 public:
298 static _Res
299 _M_invoke(const _Any_data& __functor, _ArgTypes&&... __args)
300 {
301 return (*_Base::_M_get_pointer(__functor))(
302 std::forward<_ArgTypes>(__args)...);
303 }
304 };
305
306 template<typename _Functor, typename... _ArgTypes>
307 class _Function_handler<void(_ArgTypes...), _Functor>
308 : public
309 {
310 t
311
312 pu
313 s
314
Name
Value
Type
__functor
@0x7ffffffe1b0
std::_
Type
You shouldn't need to dig around this deep in the call stack, and if you do, it should probably be a message telling you to back up a bit back to code that you actually wrote.
Debugger
GDB for "Welc
Views
Level
Function
Threads
(all)
1 nameHash
2 qMain
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

Activities Qt Creator Sun 11:30 AM

std_function.h - Qt Creator

File Edit Build Debug Analyze Tools Window Help

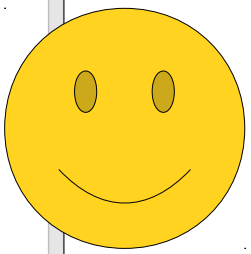
Projects

- Welcome to CS106B [master]
  - Welcome to CS106B.pro
  - Headers
  - Sources
    - lib
      - StanfordCPPLib
        - spl.cpp
      - addr2line.exe
      - addr2line64.exe
      - iconstrip.png
      - splicon-large.png
    - res
    - src
      - NameHash.cpp
  - Other files

std_function.h | _ArgTypes | Unix (LF) | Line: 302, Col: 1

Warning: The code model could not parse an included file, which might lead to incorrect code completion and highlighting, for example. Show Details Minimize

```
290
291   template<typename _Res, typename _Functor, typename... _ArgTypes>
292     class _Function_handler<_Res(_ArgTypes...), _Functor>
293     : public _Function_base::_Base_manager<_Functor>
294     {
295       typedef _Function_base::_Base_manager<_Functor> _Base;
296
297     public:
298       static _Res
299       _M_invoke(const _Any_data& __functor, _ArgTypes&&... __args)
300       {
301         return (*_Base::_M_get_pointer(__functor))(
302             std::forward<_ArgTypes>(__args)...);
303       }
304     };
305
306   template<typename _Functor, typename... _ArgTypes>
307     class _Function_handler<void(_ArgTypes...), _Functor>
308     : public
309     {
310       t
311
312      pu
313       s
314       _
```



So let's jump back to the code that we actually wrote.

Debugger | GDB for "Welc

Level	Function		
1	nameHash		
2	qMain		
3	std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&)	std_f...	302
4	std::function<int ()>::operator()() const	std_f...	706
5	QtGui::<lambda()>::operator()(void) const	spl.cpp	20981
6	std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn...	std_f...	302
7	std::function<int ()>::operator()() const	std_f...	706
8	GThreadStd::run	spl.cpp	22491
9	GThreadStd::<lambda()>::operator()(void) const	spl.cpp	22514
10	std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o...	invok...	60

Name | Value | Type

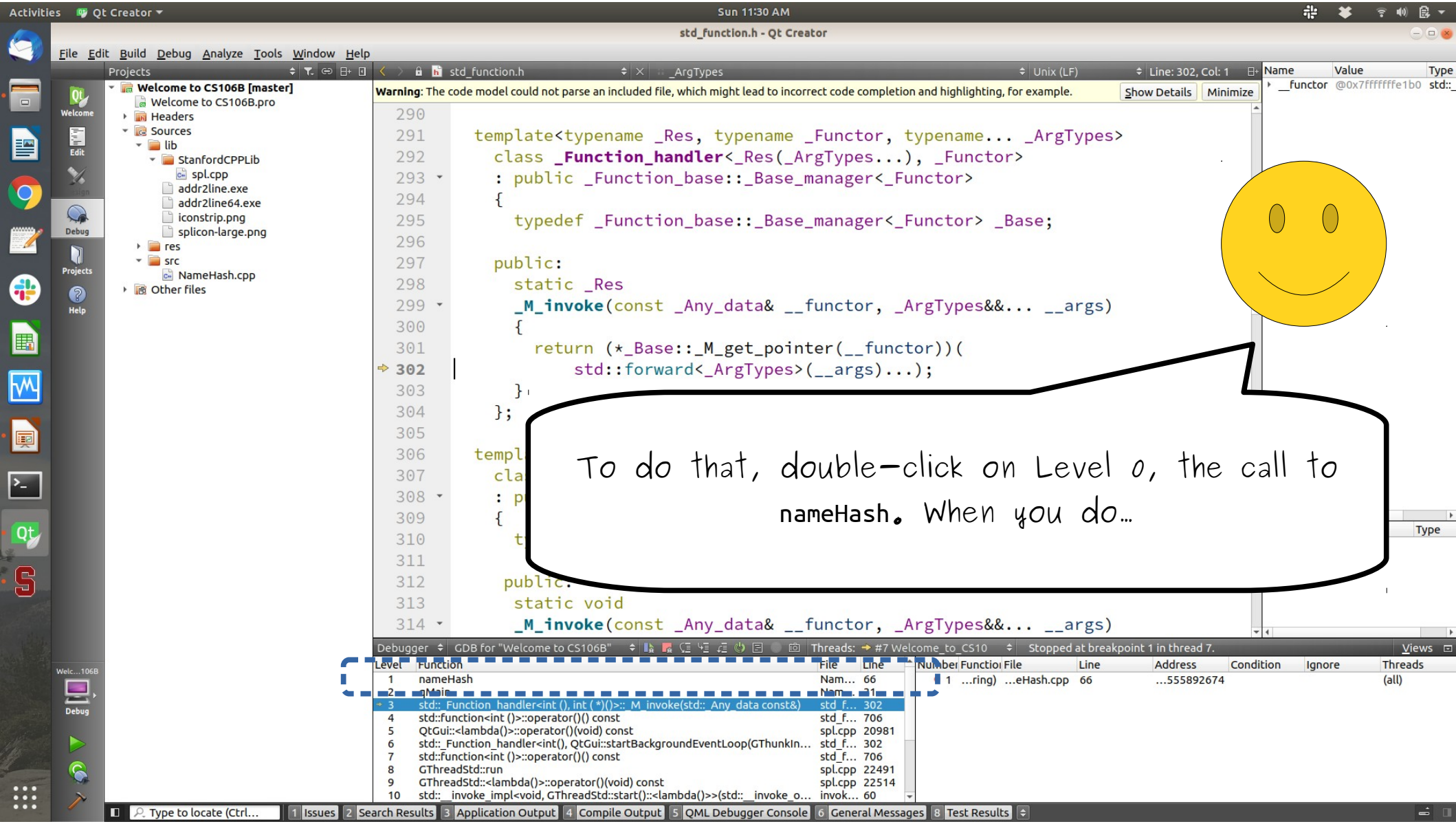
__functor @0x7fffffffe1b0 std::_

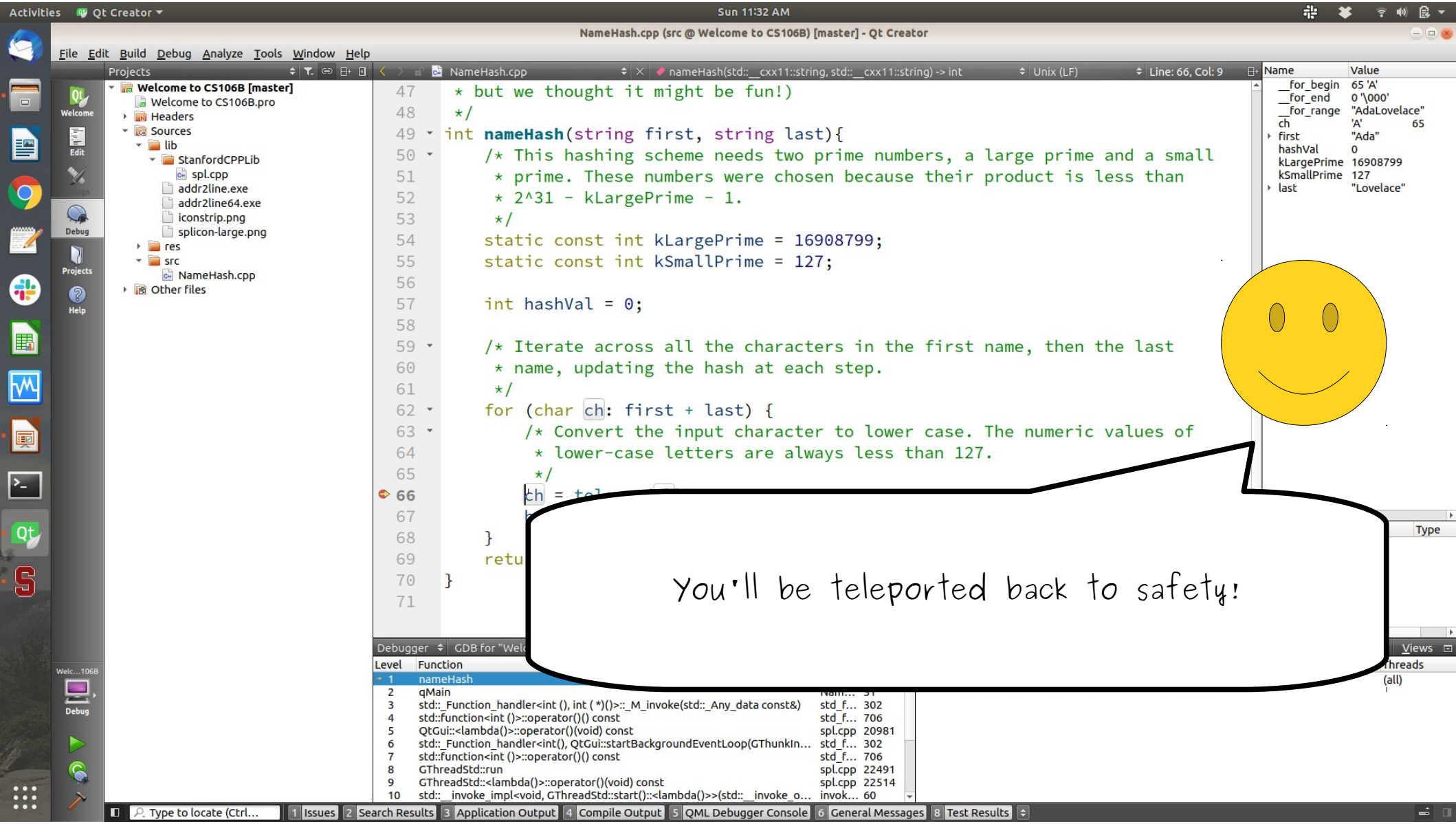
Views

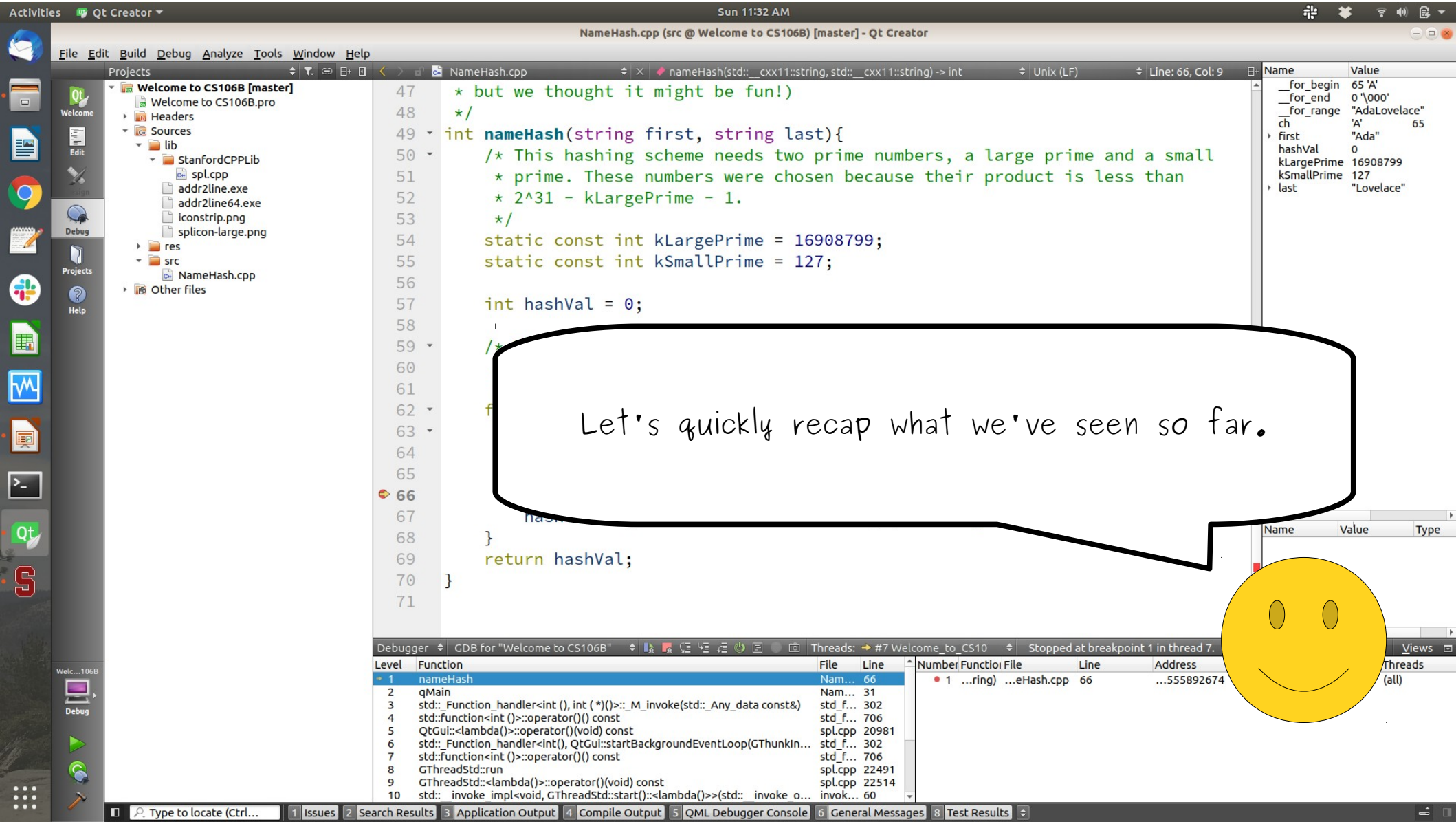
Threads

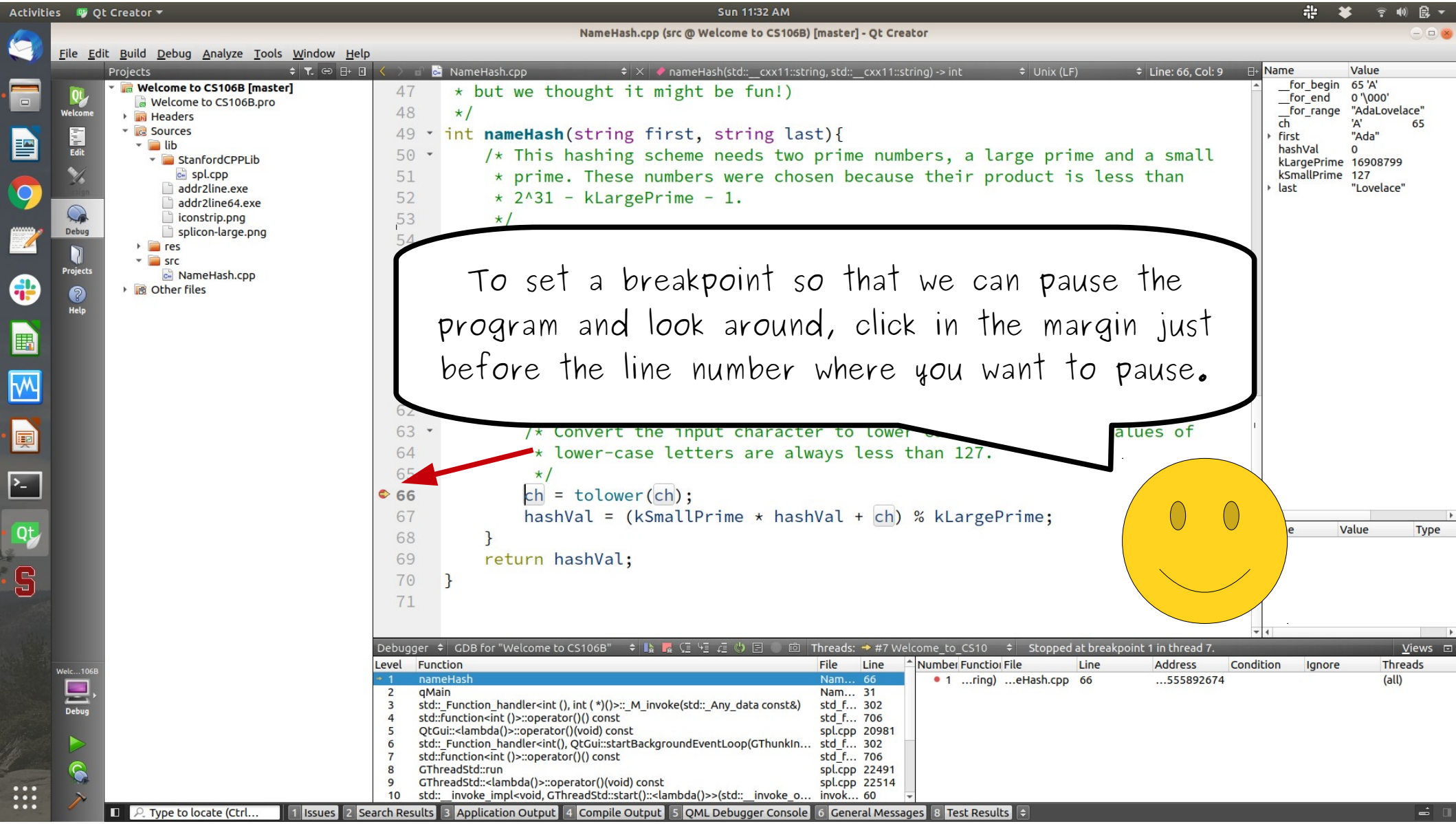
(all)

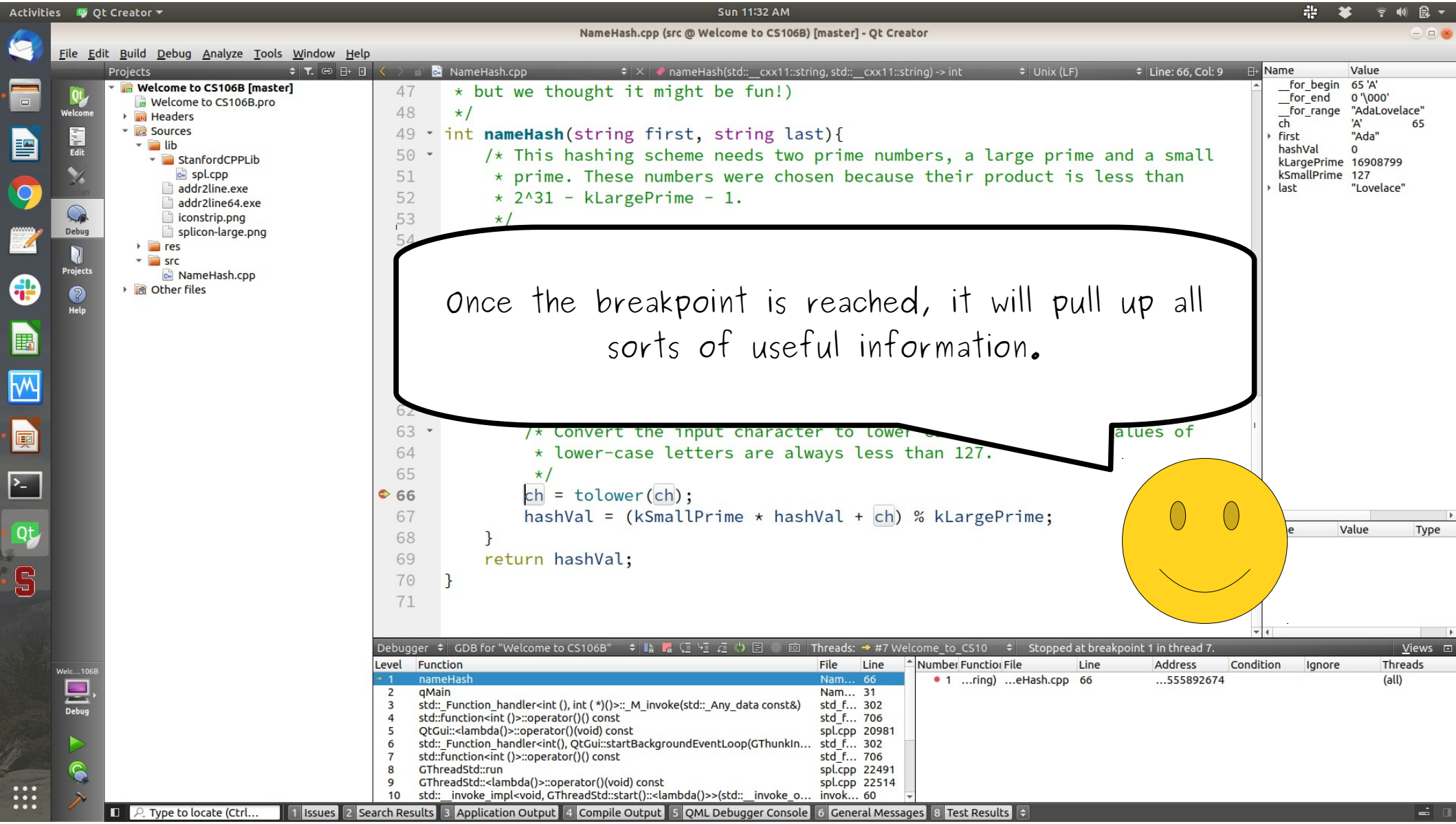
Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

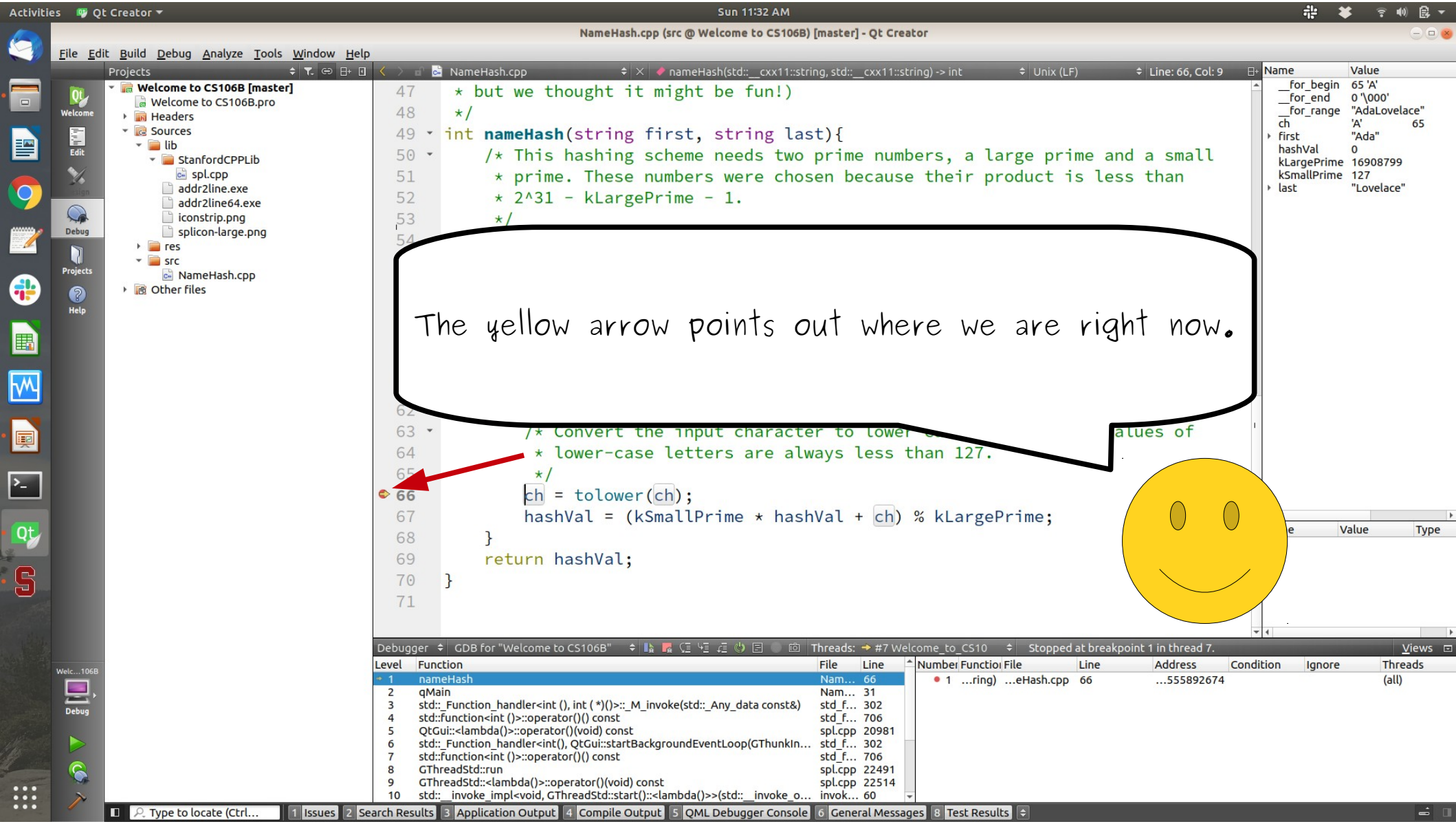

Activities
Qt Creator
Sun 11:30 AM
std_function.h - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
std_function.h
_ArgTypes
Unix (LF)
Line: 302, Col: 1
Warning: The code model could not parse an included file, which might lead to incorrect code completion and highlighting, for example.
Show Details
Minimize
290
291 template<typename _Res, typename _Functor, typename... _ArgTypes>
292 class _Function_handler<_Res(_ArgTypes...), _Functor>
293 : public _Function_base::_Base_manager<_Functor>
294 {
295 typedef _Function_base::_Base_manager<_Functor> _Base;
296
297 public:
298 static _Res
299 _M_invoke(const _Any_data& __functor, _ArgTypes&&... __args)
300 {
301 return (*_Base::_M_get_pointer(__functor))(
302 std::forward<_ArgTypes>(__args)...);
303 }
304 };
305
306 templ
307 cla
308 : p
309 {
310 t
311
312 public:
313 static void
314 _M_invoke(const _Any_data& __functor, _ArgTypes&&... __args)
Name Value Type
__functor @0x7fffffffe1b0 std::_
To do that, double-click on Level 0, the call to nameHash. When you do…
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam… 66
2 gMain Nam… 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f… 302
4 std::function<int ()>::operator()() const std_f… 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn… std_f… 302
7 std::function<int ()>::operator()() const std_f… 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o… invok… 60
Number Function File Line Address Condition Ignore Threads
1 …ring) …eHash.cpp 66 …555892674 (all)
Type
Type to locate (Ctrl…
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

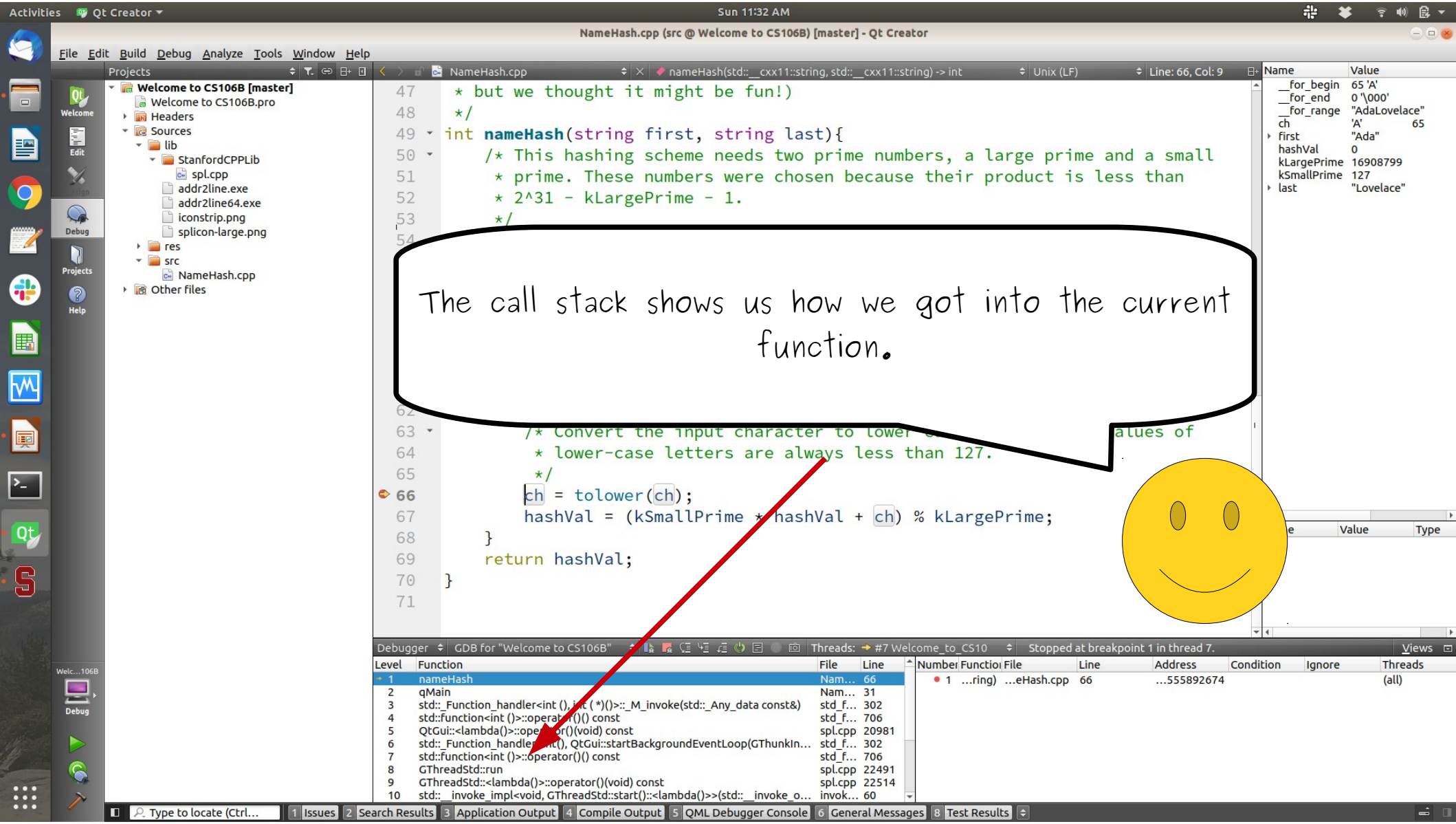

Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tol
67 h
68 }
69 retu
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
You'll be teleported back to safety!
Debugger
GDB for "Wel
Level Function
1 nameHash
2 qMain
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Views
Threads
(all)
Type
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

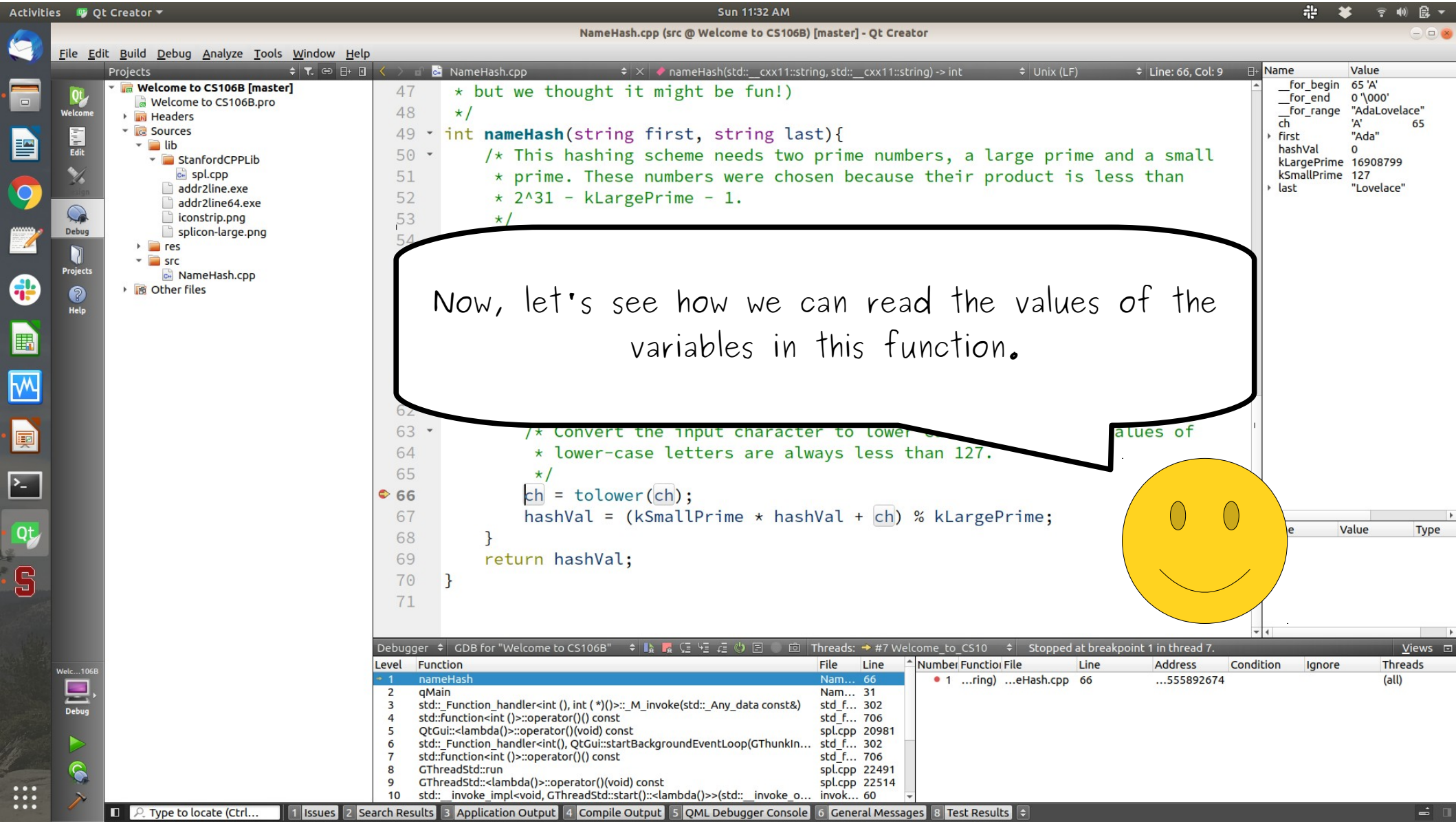

Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59
60
61
62
63
64
65
66
67
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Let's quickly recap what we've seen so far.
Name Value Type
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

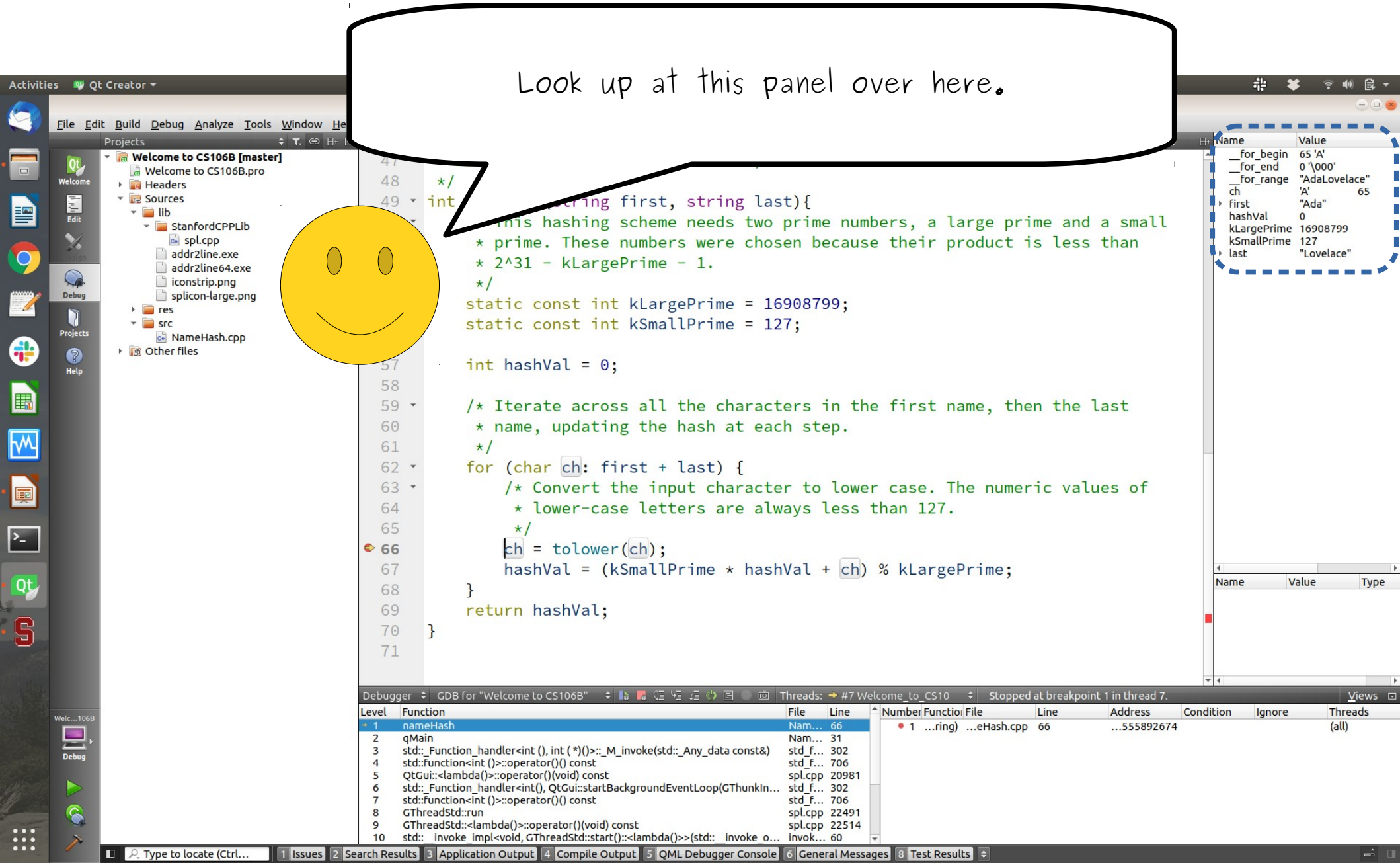

Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
63 /* Convert the input character to lower
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
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.
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Value Type
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

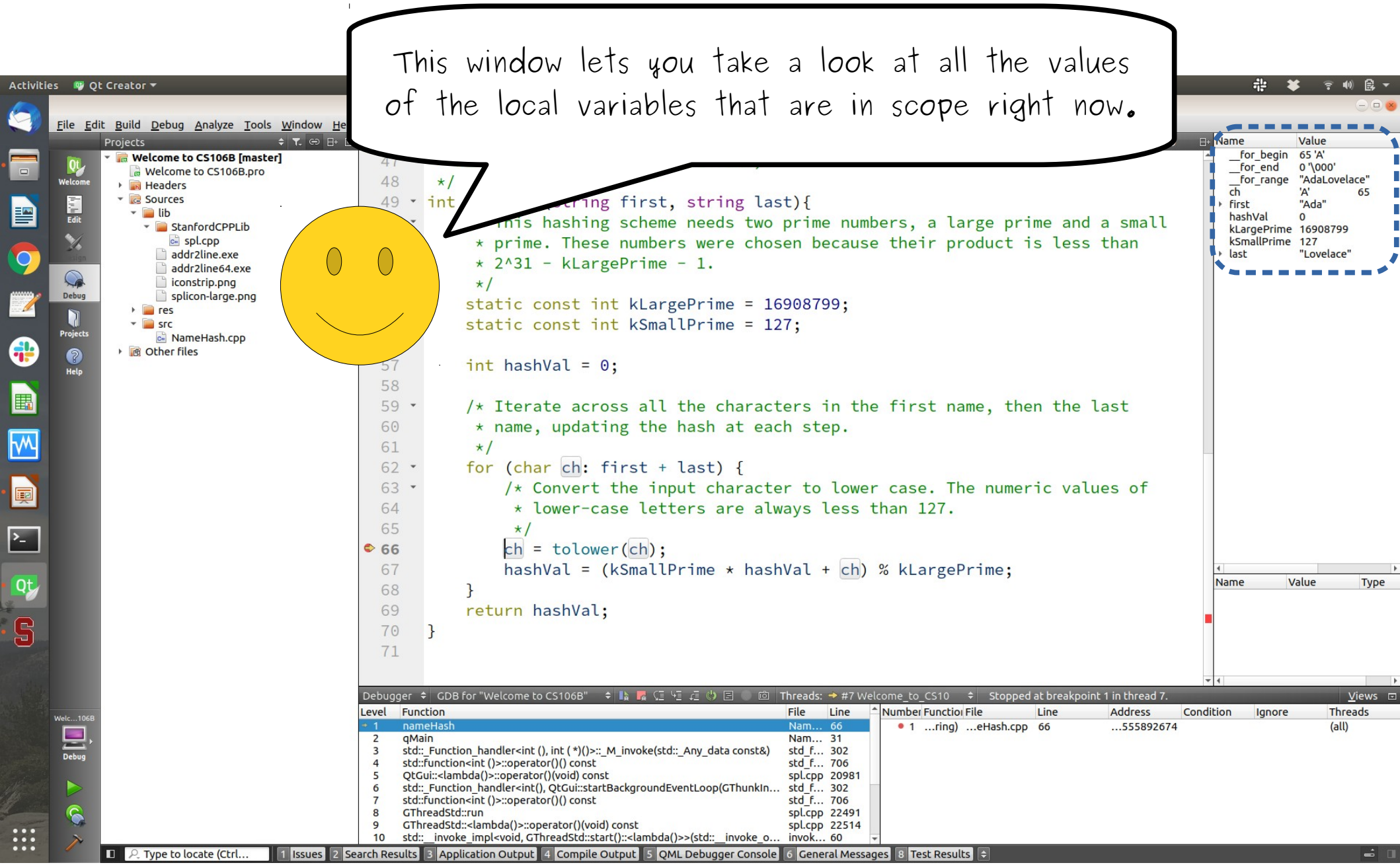

Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
63 /* Convert the input character to lower
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Once the breakpoint is reached, it will pull up all sorts of useful information.
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Value Type
Debugger GDB for "Welcome to CS106B" Threads: #7 Welcome_to_CS10 Stopped at breakpoint 1 in thread 7. Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results


Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54
62
63 /* Convert the input character to lower
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
The yellow arrow points out where we are right now.
Name
Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Value
Type
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

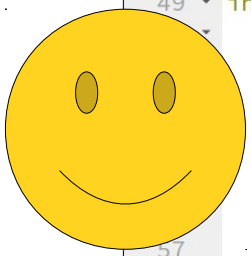

Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
63 /* Convert the input character to lower
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
The call stack shows us how we got into the current function.
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Value Type
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<void (), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results


Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54
62
63 /* Convert the input character to lower
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Now, let's see how we can read the values of the variables in this function.
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Value Type
Debugger GDB for "Welcome to CS106B" Threads: #7 Welcome_to_CS10 Stopped at breakpoint 1 in thread 7. Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results


Look up at this panel over here.
Activities
Qt Creator
File Edit Build Debug Analyze Tools Window
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
Welcome
Edit
Debug
Projects
Help
*/
int ...tring first, string last){
* This hashing scheme needs two prime numbers, a large prime and a small
* prime. These numbers were chosen because their product is less than
* 2^31 - kLargePrime - 1.
*/
static const int kLargePrime = 16908799;
static const int kSmallPrime = 127;
int hashVal = 0;
/* Iterate across all the characters in the first name, then the last
* name, updating the hash at each step.
*/
for (char ch: first + last) {
/* Convert the input character to lower case. The numeric values of
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

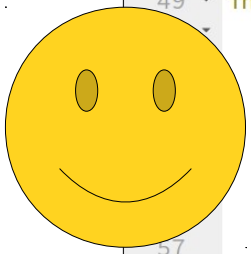

This window lets you take a look at all the values of the local variables that are in scope right now.
Activities
Qt Creator
File Edit Build Debug Analyze Tools Window
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
Welcome
Edit
Debug
Projects
Help
*/
int ...tring first, string last){
* This hashing scheme needs two prime numbers, a large prime and a small
* prime. These numbers were chosen because their product is less than
* 2^31 - kLargePrime - 1.
*/
static const int kLargePrime = 16908799;
static const int kSmallPrime = 127;
int hashVal = 0;
/* Iterate across all the characters in the first name, then the last
* name, updating the hash at each step.
*/
for (char ch: first + last) {
/* Convert the input character to lower case. The numeric values of
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
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`.



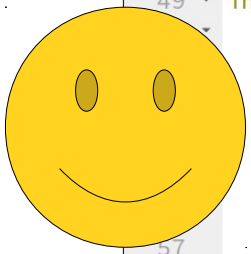
```
 */
int ...tring first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

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

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



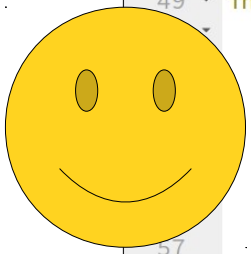
```
 */
int                    string first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

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

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



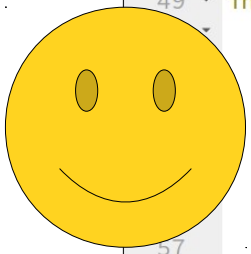
```
 */
int                    tring first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

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

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

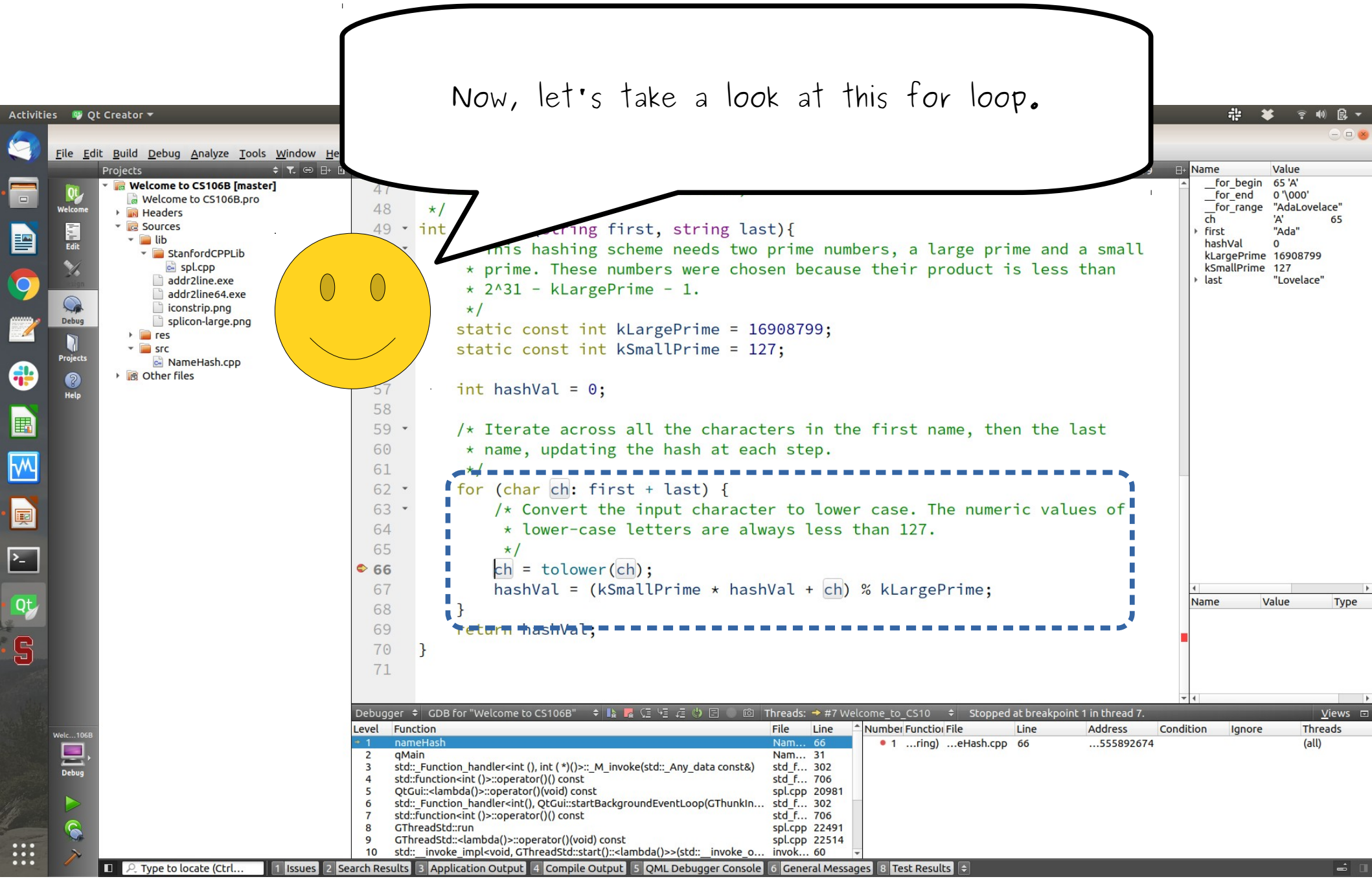


```
 */
int                    string first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

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

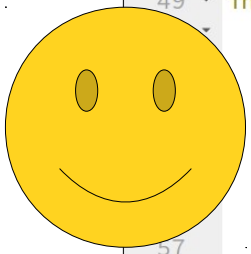
As we walk through the program one step at a time, we'll see these values change.



```
 */
int                    tring first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

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

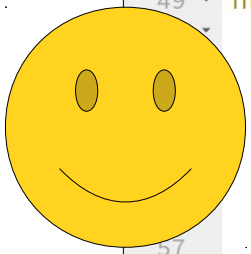

Now, let's take a look at this for loop.
Activities
Qt Creator
File Edit Build Debug Analyze Tools Window
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
*/
int ring first, string last){
this hashing scheme needs two prime numbers, a large prime and a small
* prime. These numbers were chosen because their product is less than
* 2^31 - kLargePrime - 1.
*/
static const int kLargePrime = 16908799;
static const int kSmallPrime = 127;
int hashVal = 0;
/* Iterate across all the characters in the first name, then the last
* name, updating the hash at each step.
*/
for (char ch: first + last) {
/* Convert the input character to lower case. The numeric values of
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
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."



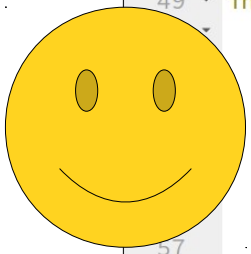
```
 */
int                 tring first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

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

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



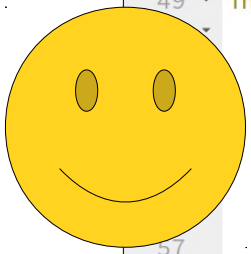
```
 */
int                    string first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

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

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.



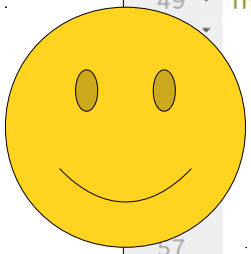
```
 */
int                    tring first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first name, then the last
     * name, updating the hash at each step.
     */
    for (char ch: first + last) {
        /* Convert the input character to lower case. The numeric values of
         * 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.

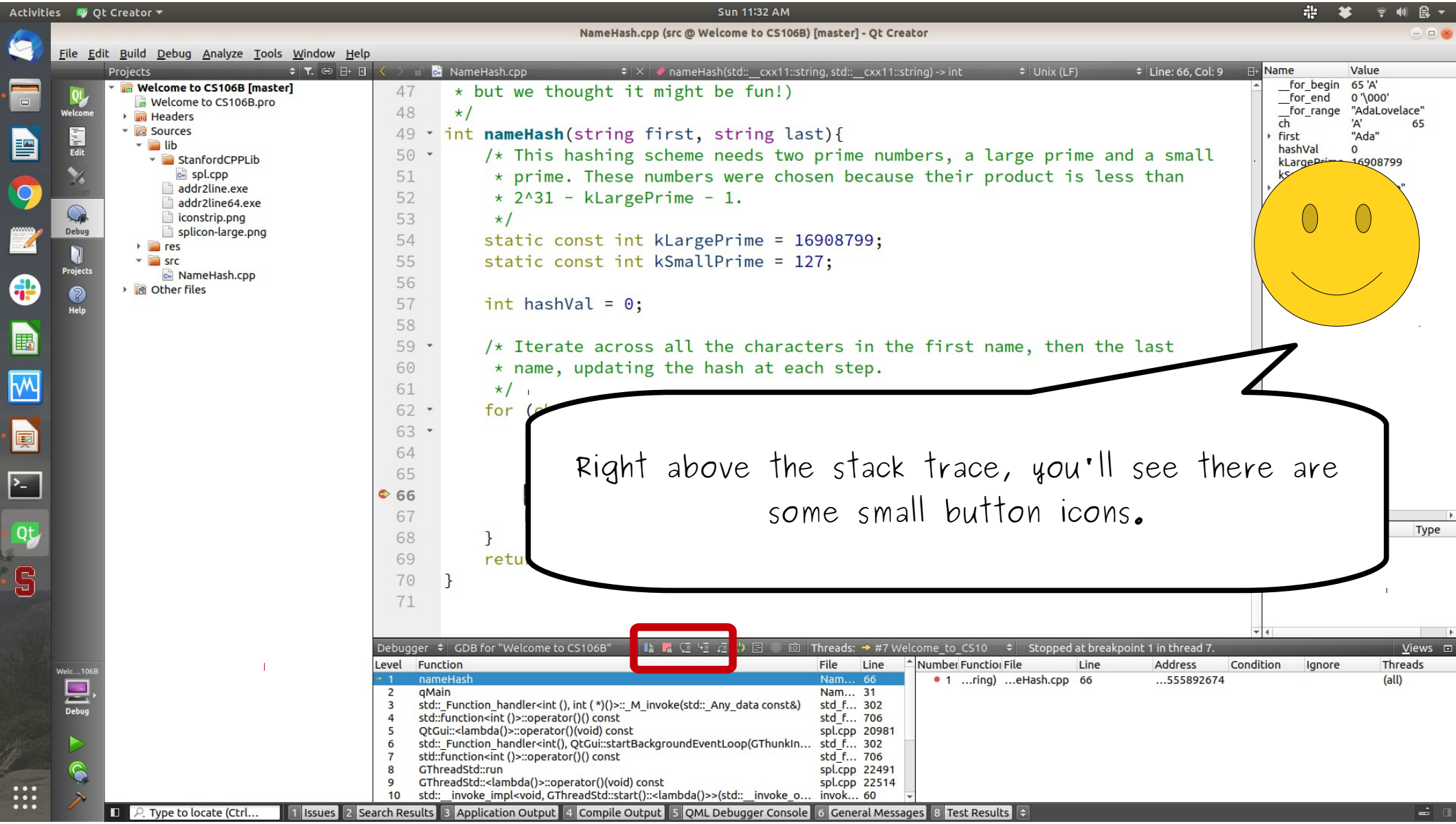


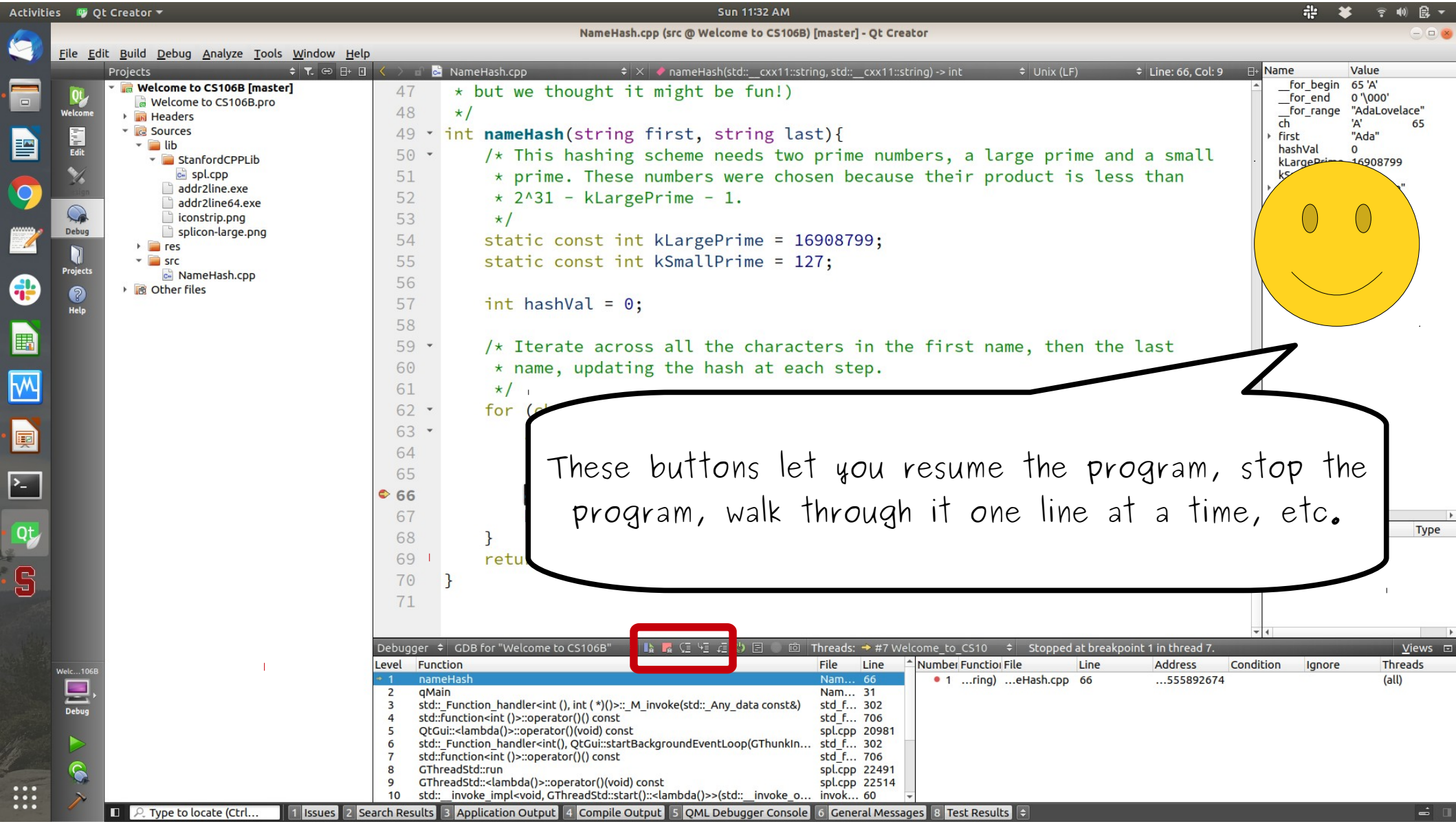
```
 */
int                    string first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

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

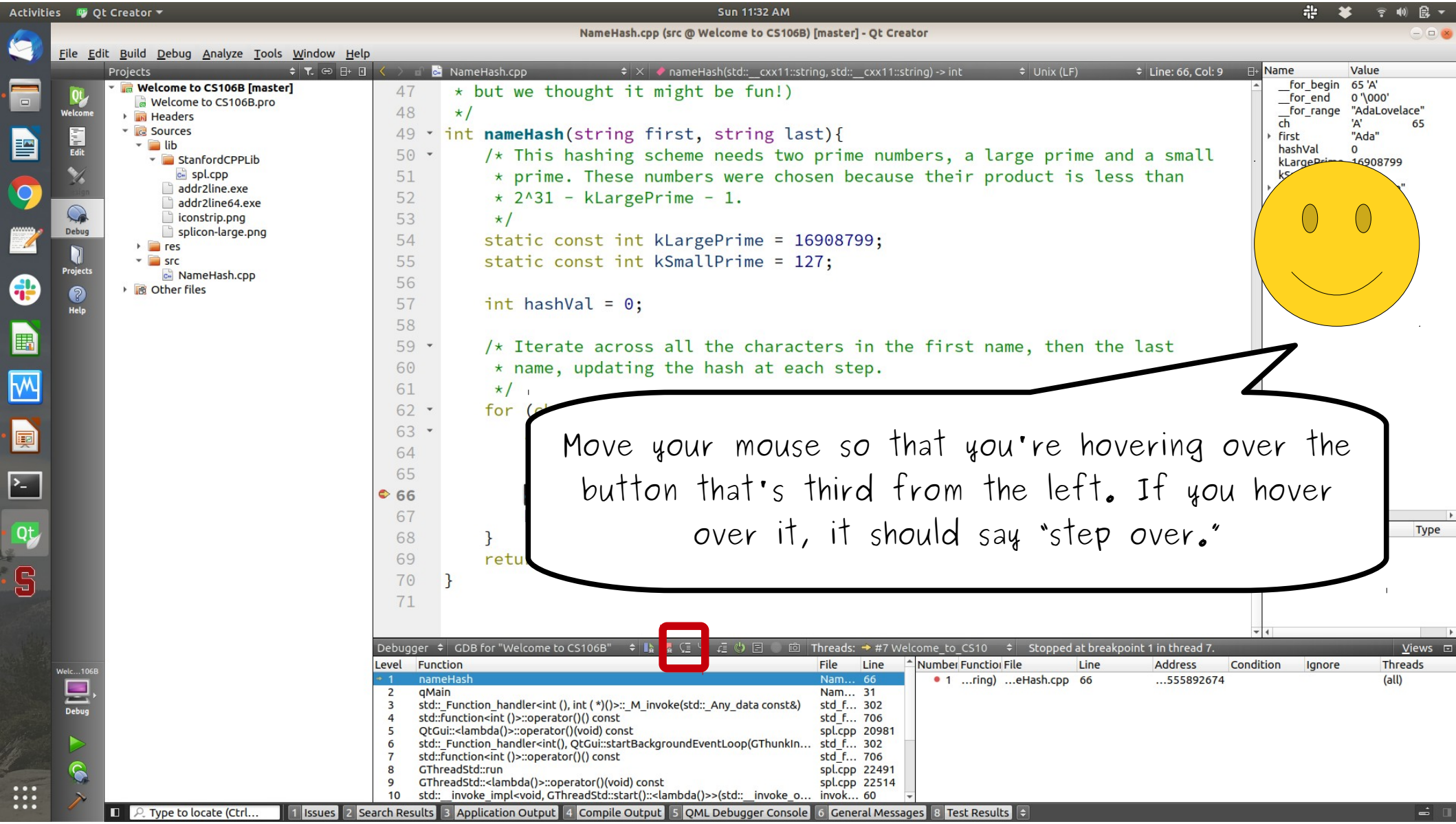
Now, let's do something really cool - we're going to run this program one line at a time, watching what happens at each step!

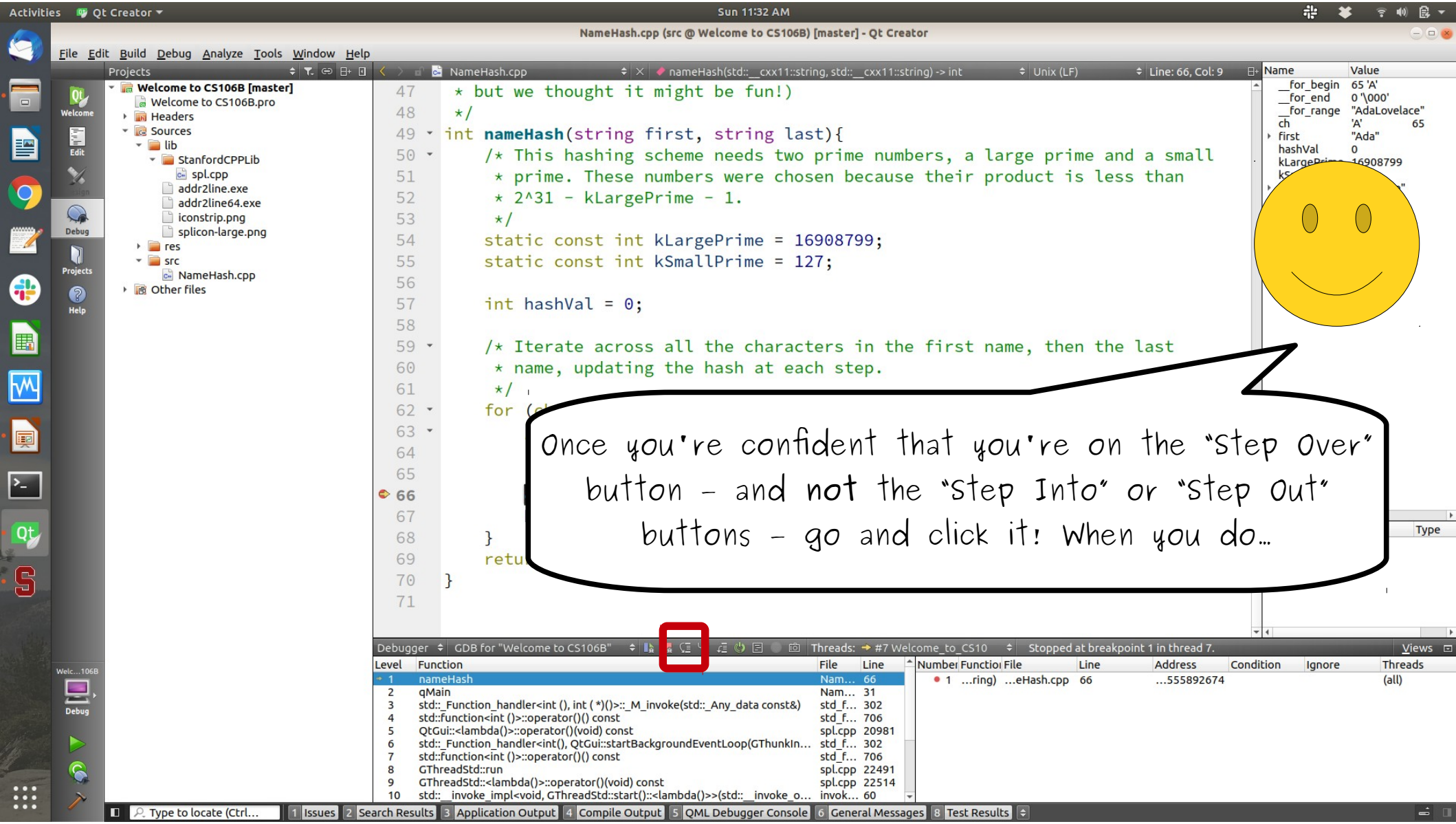


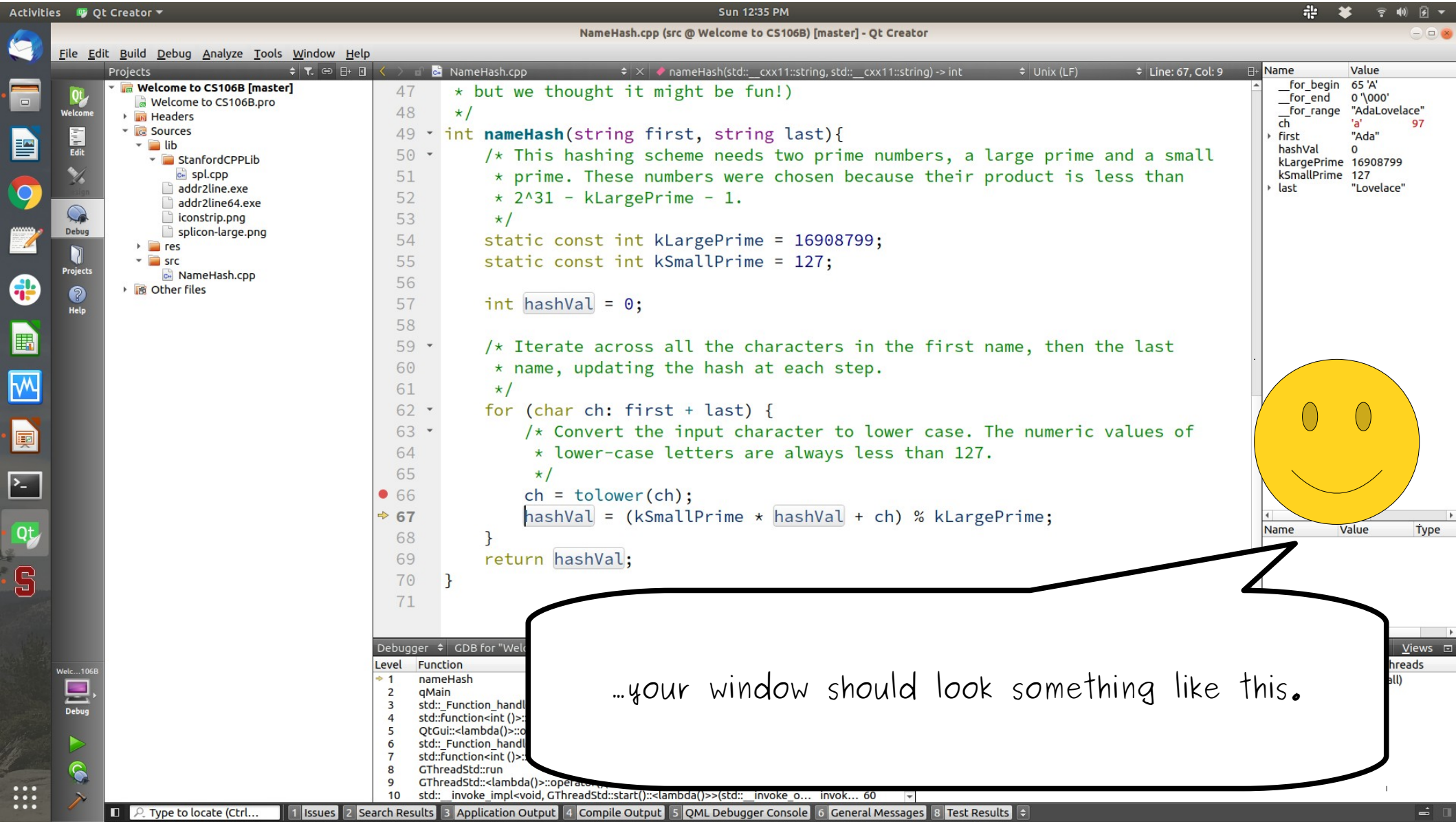
```
 */
int                    tring first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

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

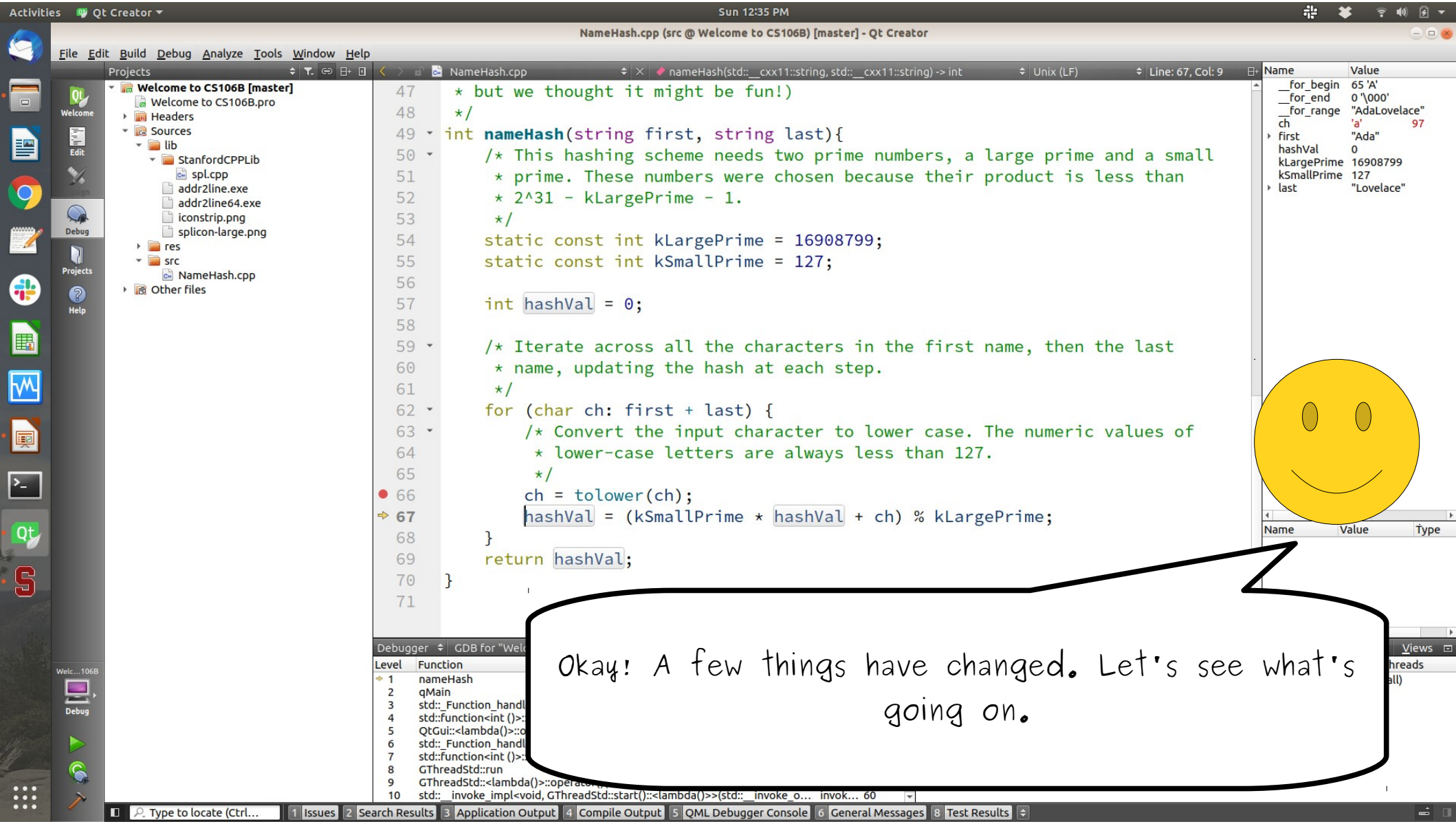

Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name
Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
Type
Right above the stack trace, you'll see there are some small button icons.
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

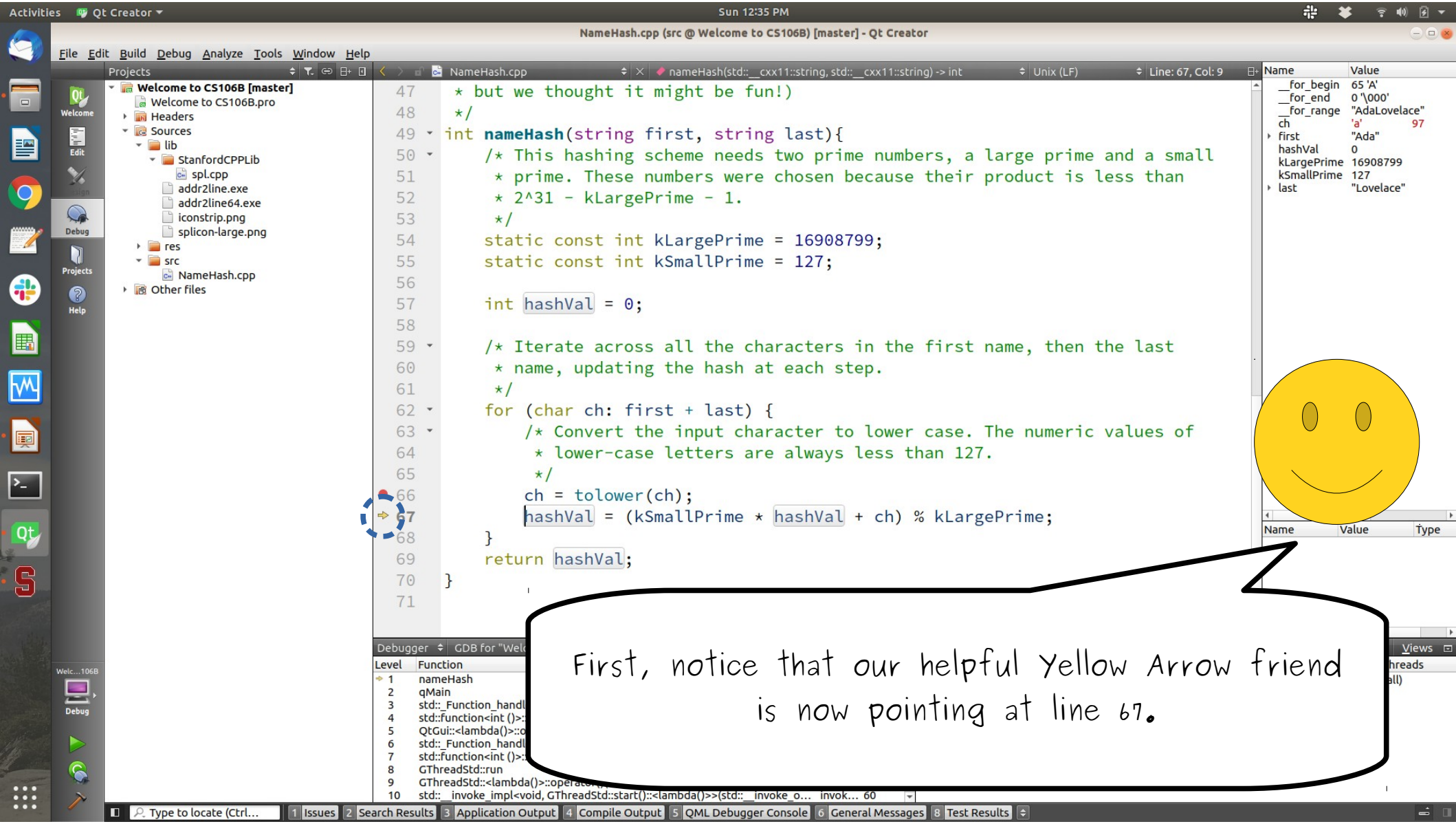

Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name
Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
Type
These buttons let you resume the program, stop the program, walk through it one line at a time, etc.
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results
Welcome
Edit
Debug
Projects
Help
Welc...106B
Debug


Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name
Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrim 16908799
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."
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results


Activities
Qt Creator
Sun 11:32 AM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name
Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
Type
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...
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

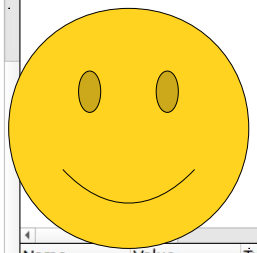

Activities
Qt Creator
Sun 12:35 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 67, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
Level Function
1 nameHash
2 qMain
Views
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results
…your window should look something like this.


Activities
Qt Creator
Sun 12:35 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 67, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
Level Function
1 nameHash
2 qMain
3 std::_Function_handl
4 std::function<int ()>:
5 QtGui::<lambda()>::o
6 std::_Function_handl
7 std::function<int ()>:
8 GThreadStd::run
9 GThreadStd::<lambda()>::operator
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Okay! A few things have changed. Let's see what's going on.
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

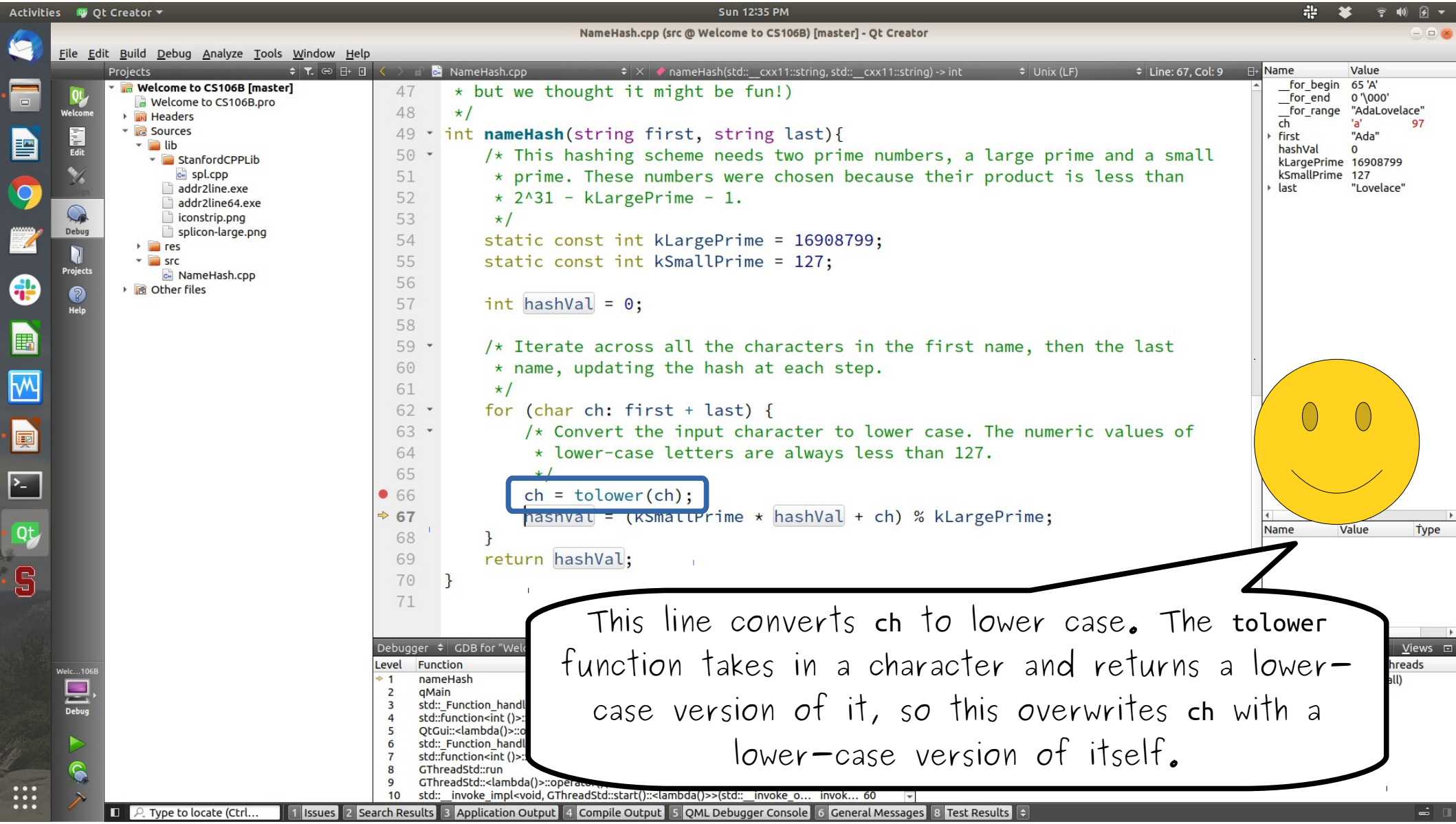

Activities
Qt Creator
Sun 12:35 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 67, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
Level Function
1 nameHash
2 qMain
3 std::_Function_handl
4 std::function<int ()>:
5 QtGui::<lambda()>::o
6 std::_Function_handl
7 std::function<int ()>:
8 GThreadStd::run
9 GThreadStd::<lambda()>::operator
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Views
Debug
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results
First, notice that our helpful Yellow Arrow friend is now pointing at line 67.


Activities
Qt Creator
Sun 12:35 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 67, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
Level Function
1 nameHash
2 qMain
3 std::_Function_handl
4 std::function<int ()>:
5 QtGui::<lambda()>::o
6 std::_Function_handl
7 std::function<int ()>:
8 GThreadStd::run
9 GThreadStd::<lambda()>::operator()
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Views
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results
We're now at the line right after the one where we stopped. You just ran a single line of the program! Pretty cool!

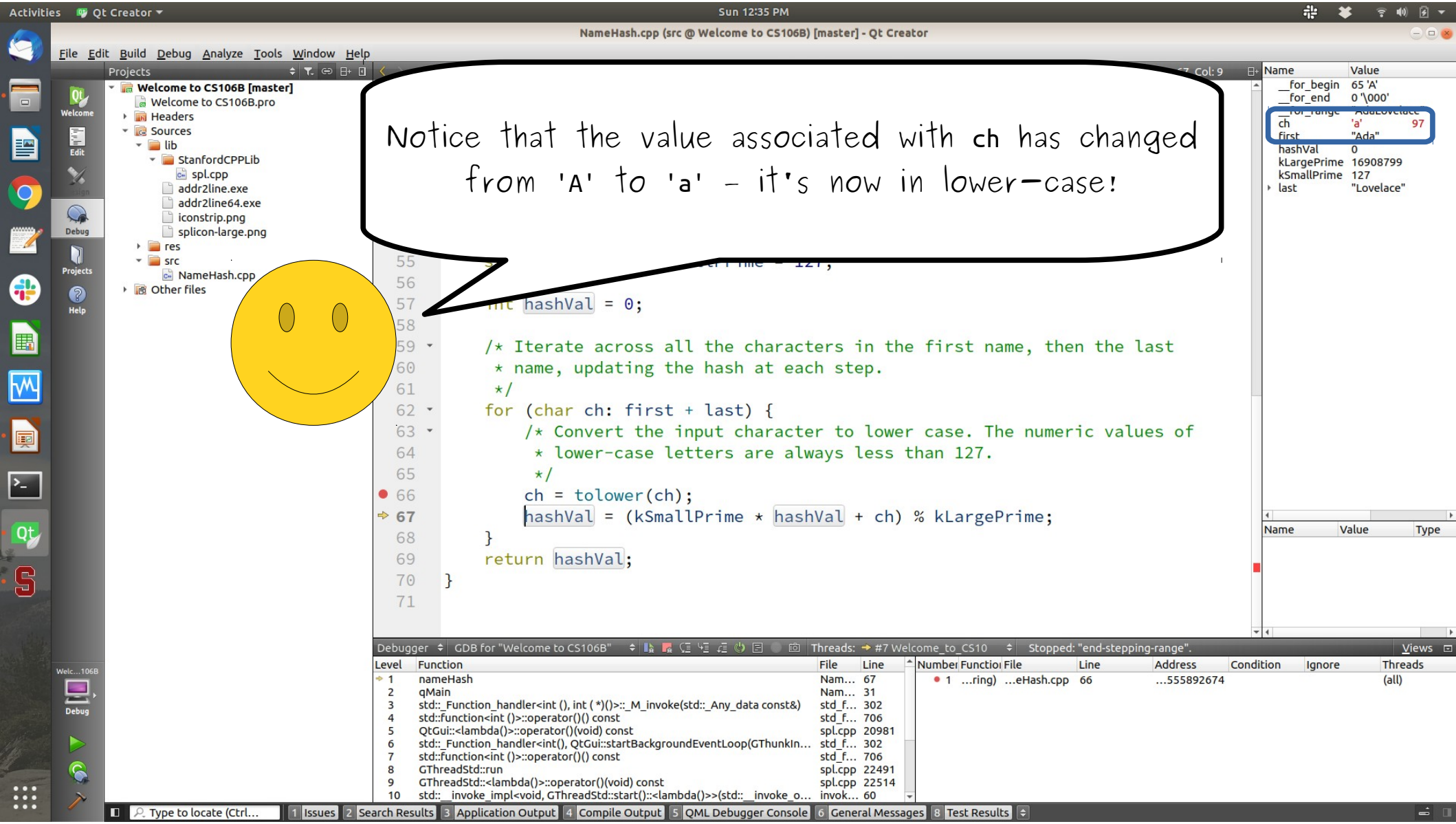
```
 * but we thought it might be fun!)
 */
int nameHash(string first, string last){
    /* This hashing scheme needs two prime numbers, a large prime and a small
     * prime. These numbers were chosen because their product is less than
     * 2^31 - kLargePrime - 1.
     */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

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

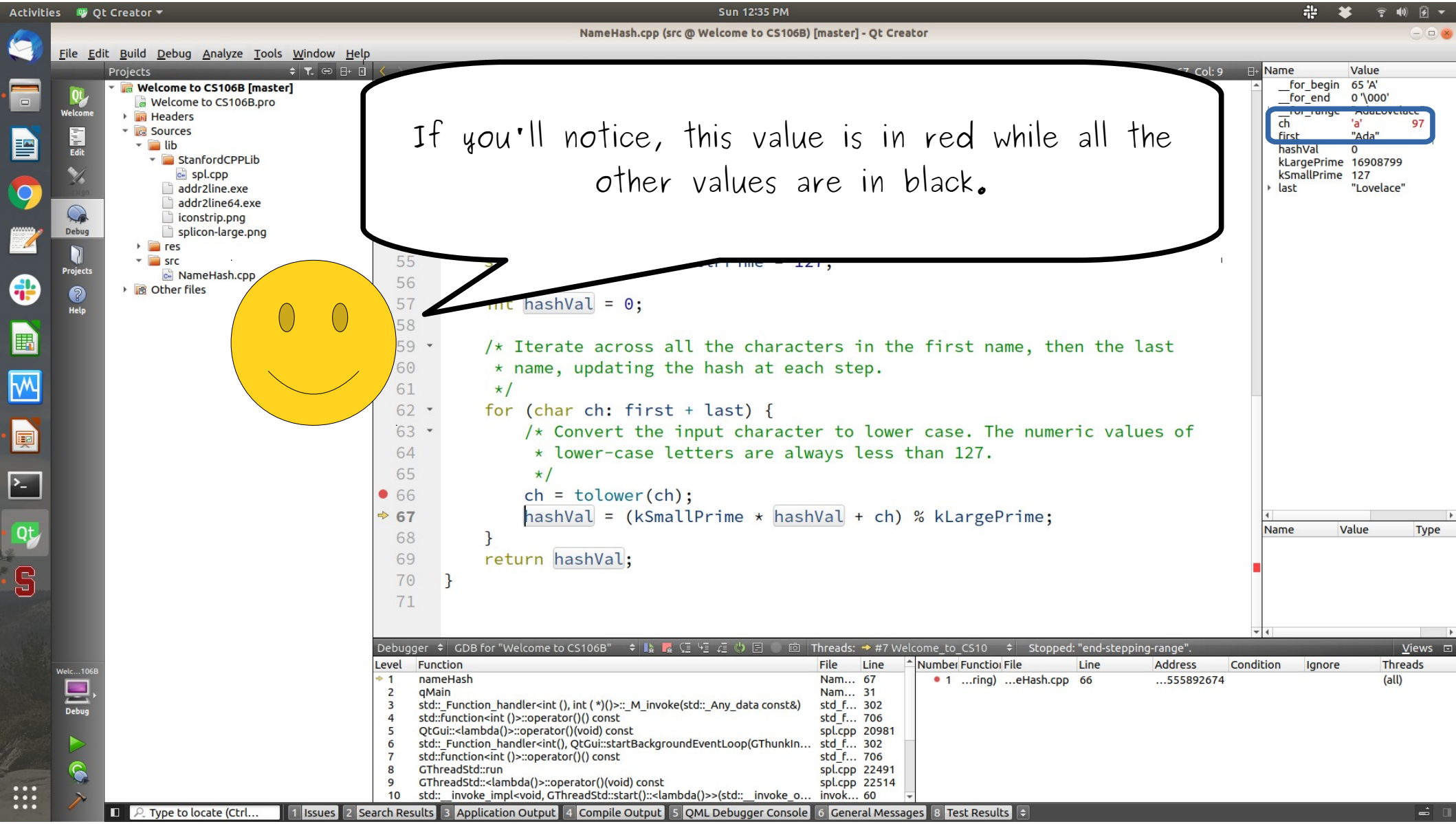


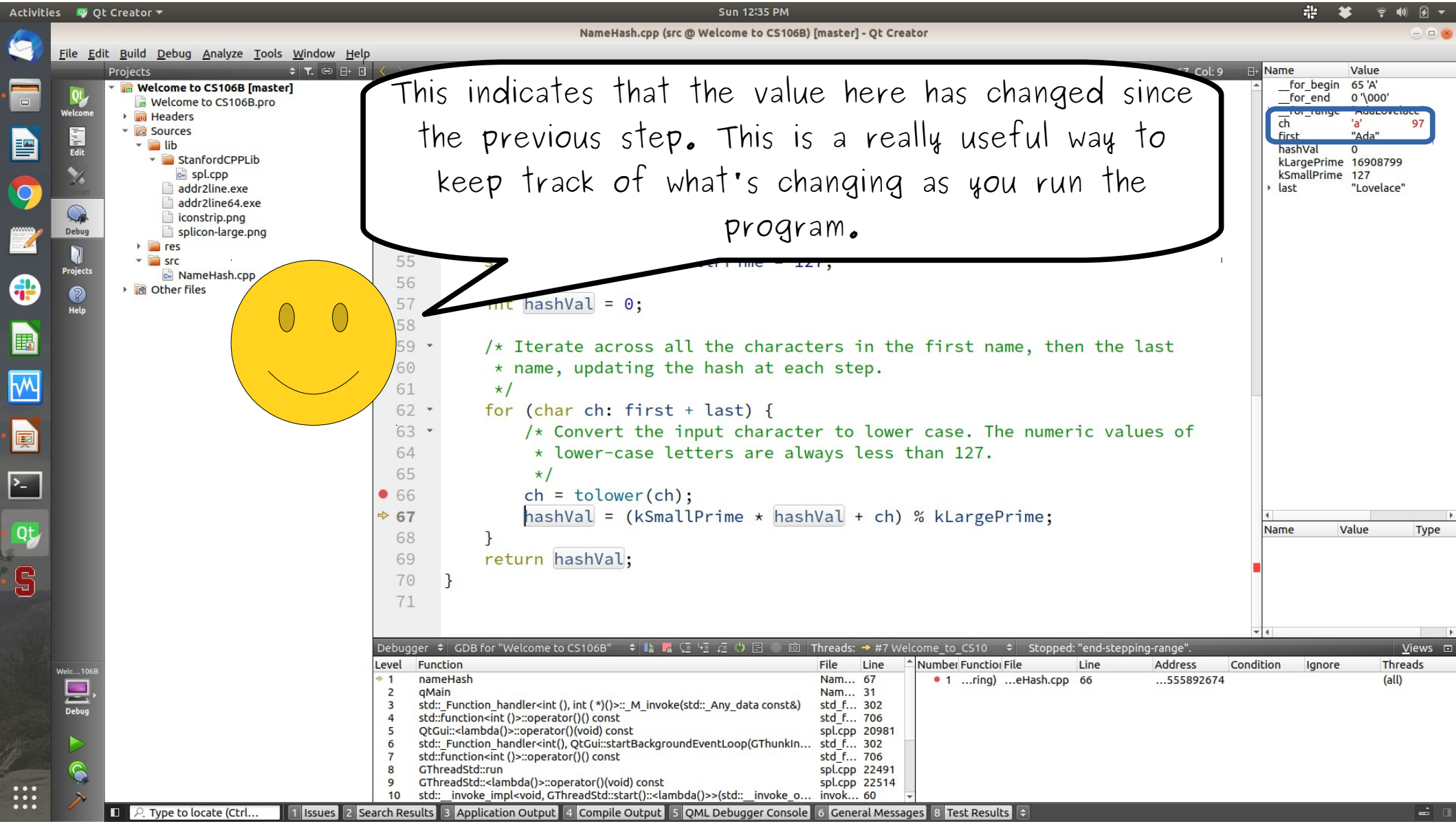
so what did that line of code do?

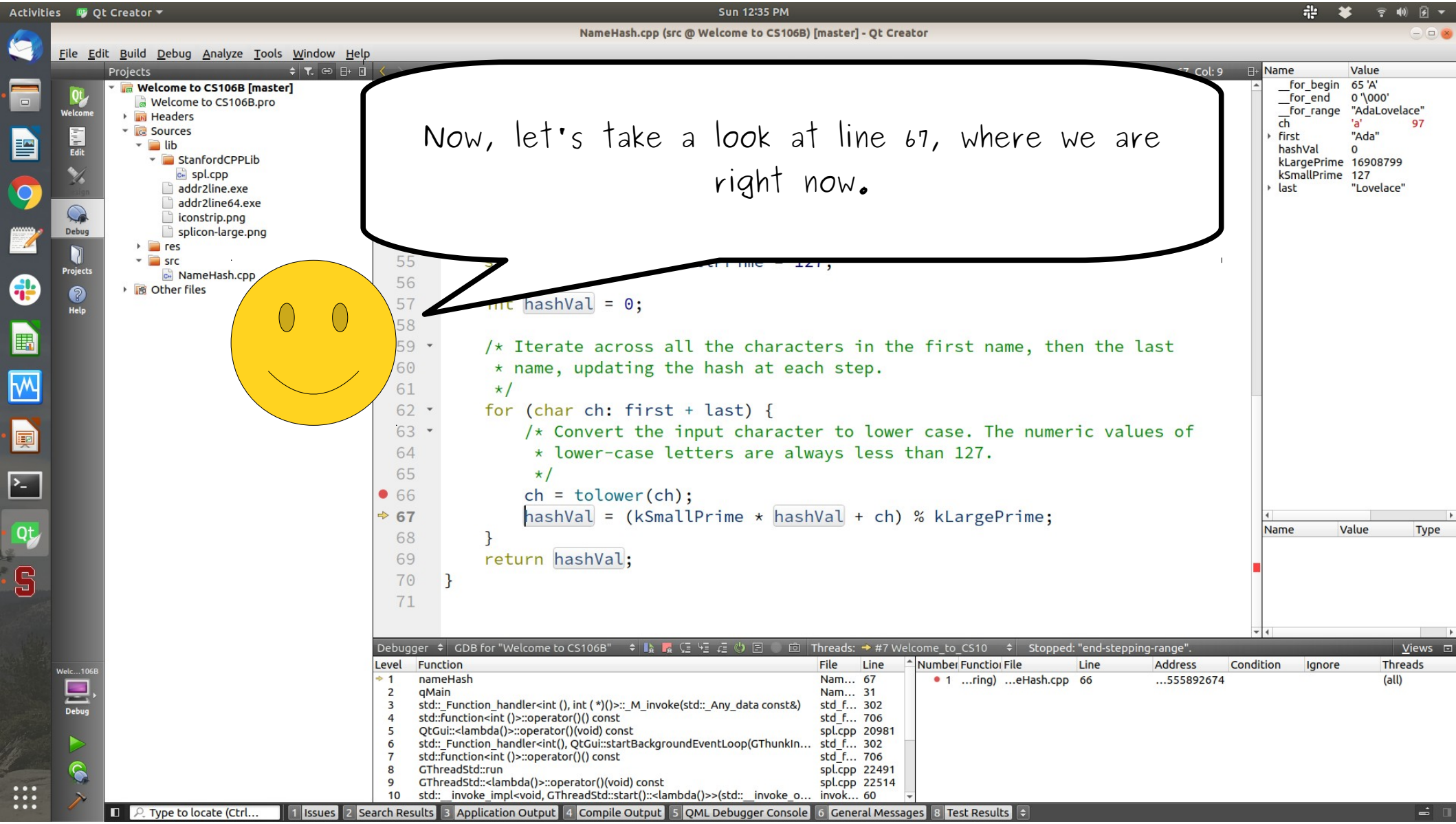

Activities
Qt Creator
Sun 12:35 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 67, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger
Level Function
1 nameHash
2 qMain
Views
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results
This line converts ch to lower case. The tolower function takes in a character and returns a lower-case version of it, so this overwrites ch with a lower-case version of itself.


You can actually see this by looking at the values panel over on the side!
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
int hashVal = 0;
/* Iterate across all the characters in the first name, then the last
* name, updating the hash at each step.
*/
for (char ch: first + last) {
/* Convert the input character to lower case. The numeric values of
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"


Activities
Qt Creator
Sun 12:35 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
Notice that the value associated with ch has changed from 'A' to 'a' - it's now in lower-case!
55
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger GDB for "Welcome to CS106B" Threads: #7 Welcome_to_CS10 Stopped: "end-stepping-range". Views
Level Function File Line
1 nameHash Nam... 67
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

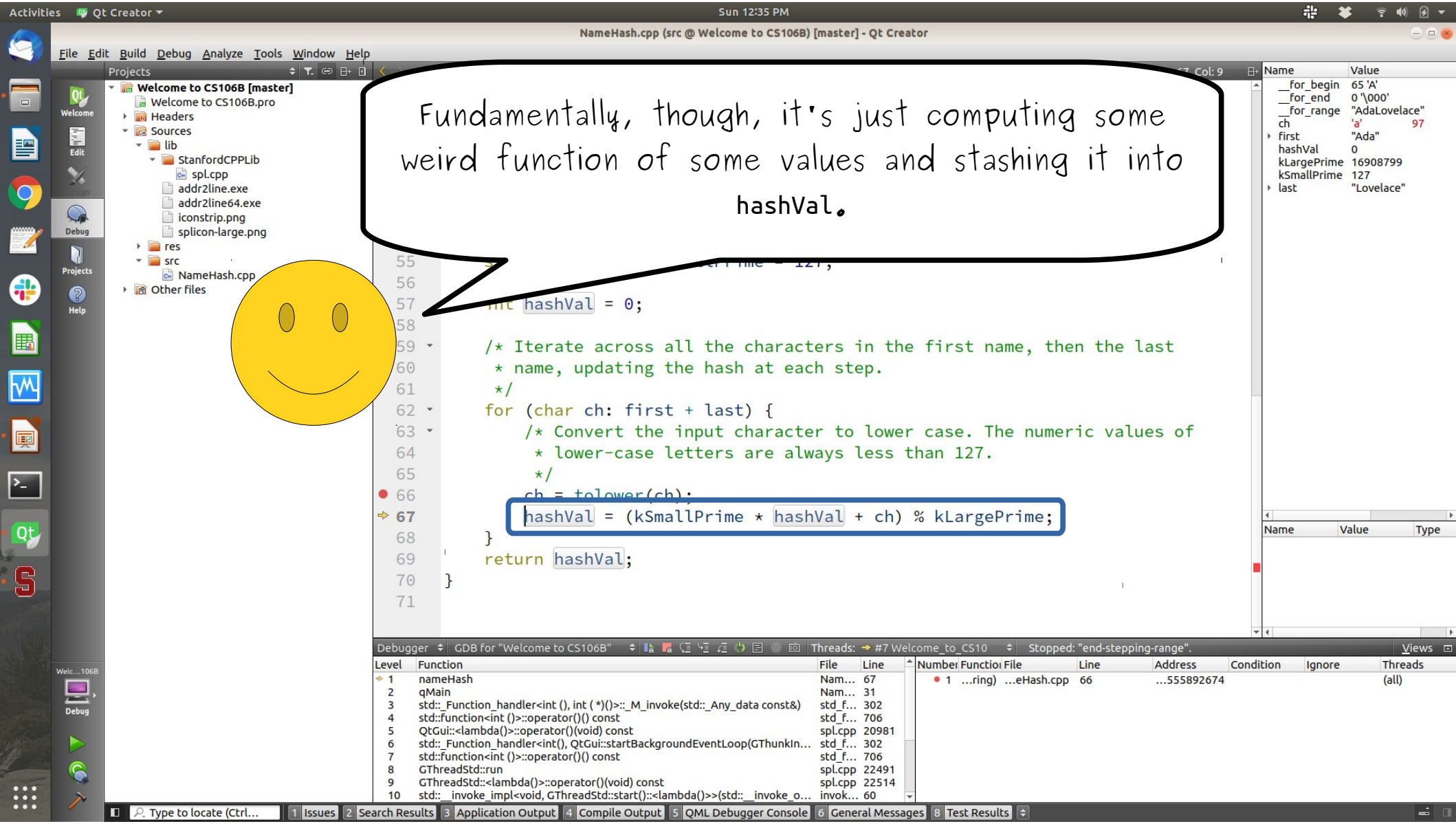

If you'll notice, this value is in red while all the other values are in black.
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
int hashVal = 0;
/* Iterate across all the characters in the first name, then the last
* name, updating the hash at each step.
*/
for (char ch: first + last) {
/* Convert the input character to lower case. The numeric values of
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"

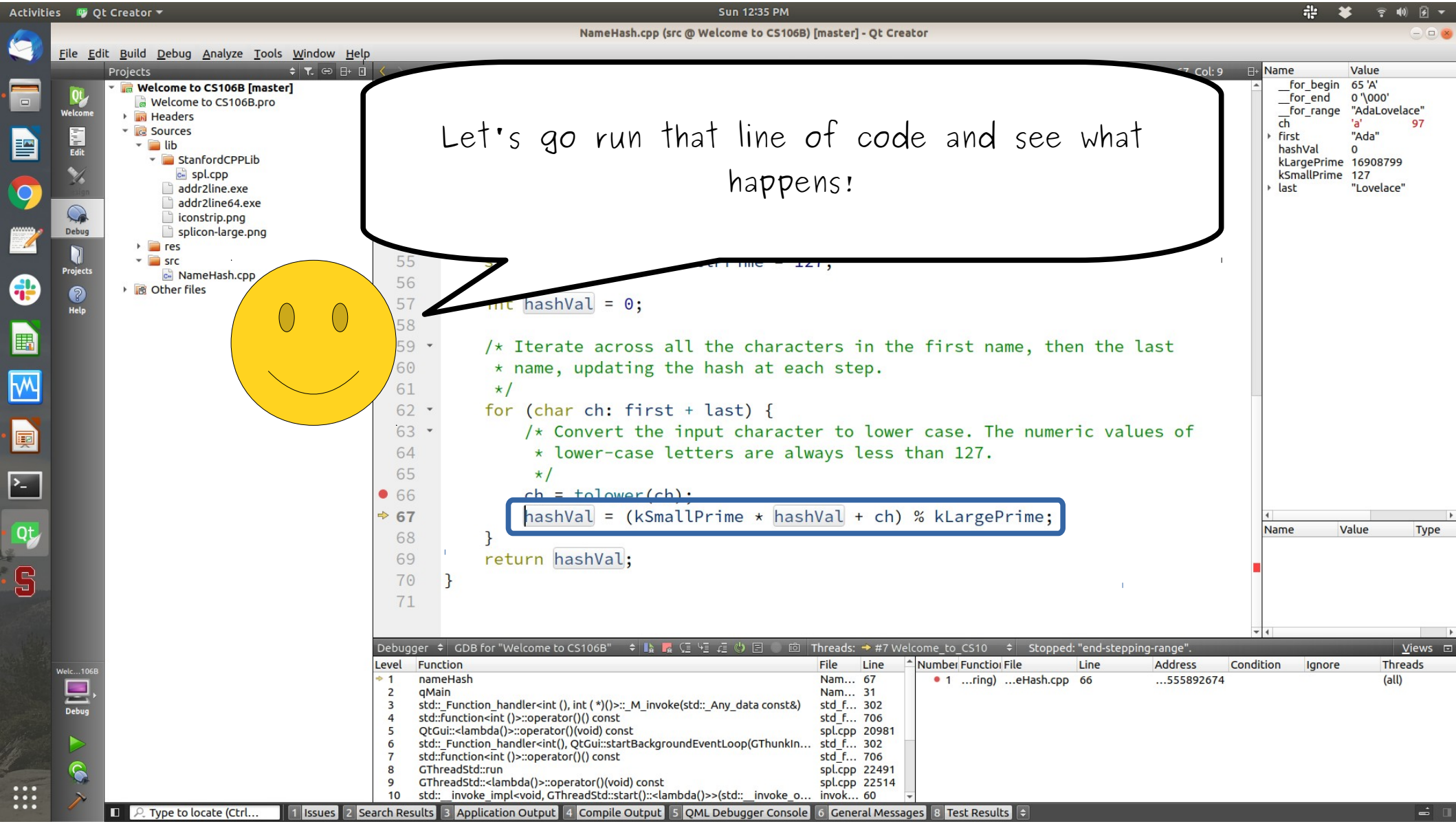

This indicates that the value here has changed since the previous step. This is a really useful way to keep track of what's changing as you run the program.
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
int hashVal = 0;
/* Iterate across all the characters in the first name, then the last
* name, updating the hash at each step.
*/
for (char ch: first + last) {
/* Convert the input character to lower case. The numeric values of
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"

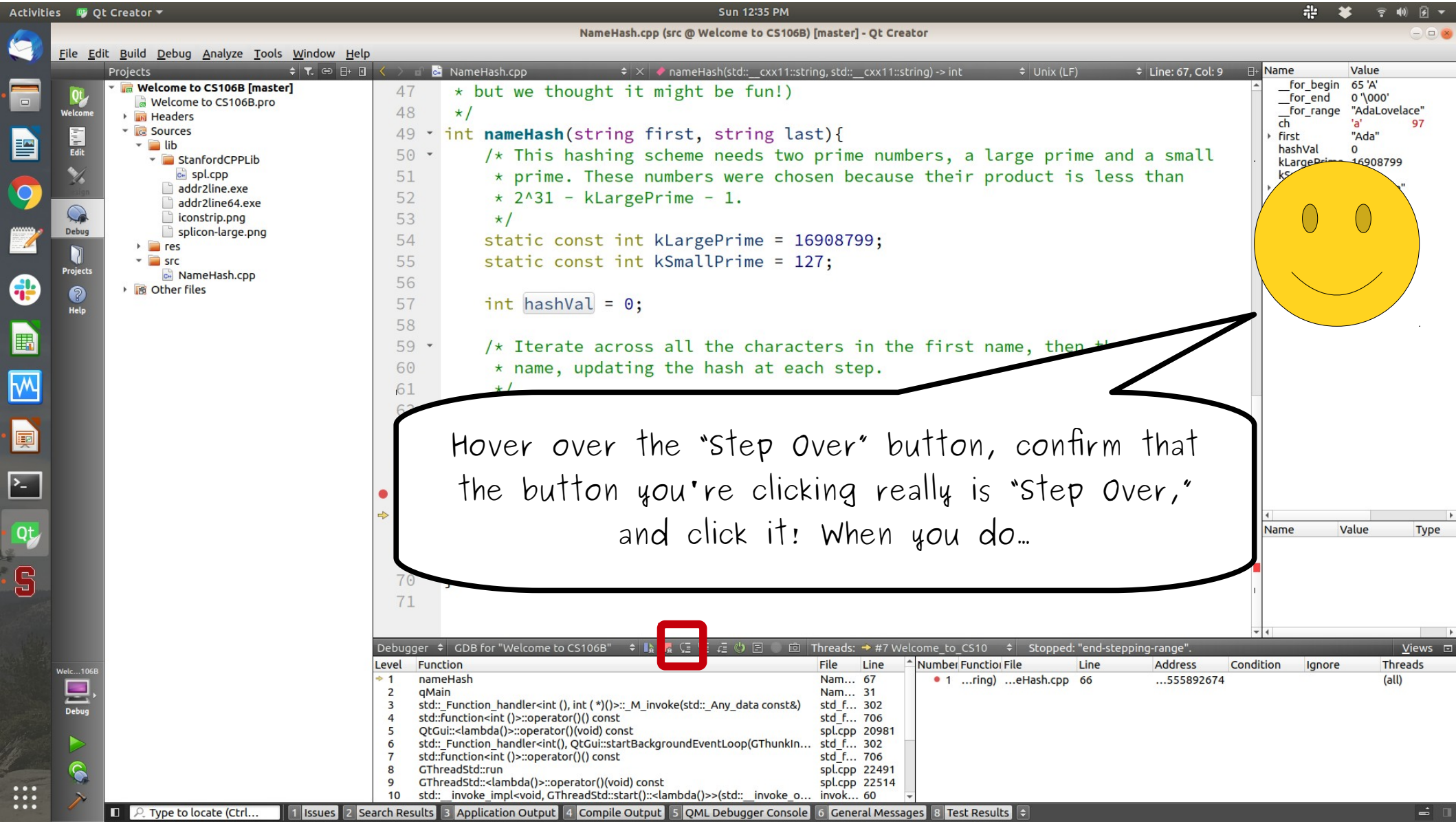

Activities
Qt Creator
Sun 12:35 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
Now, let's take a look at line 67, where we are right now.
55
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Name Value Type
Debugger GDB for "Welcome to CS106B" Threads: #7 Welcome_to_CS10 Stopped: "end-stepping-range". Views
Level Function File Line
1 nameHash Nam... 67
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

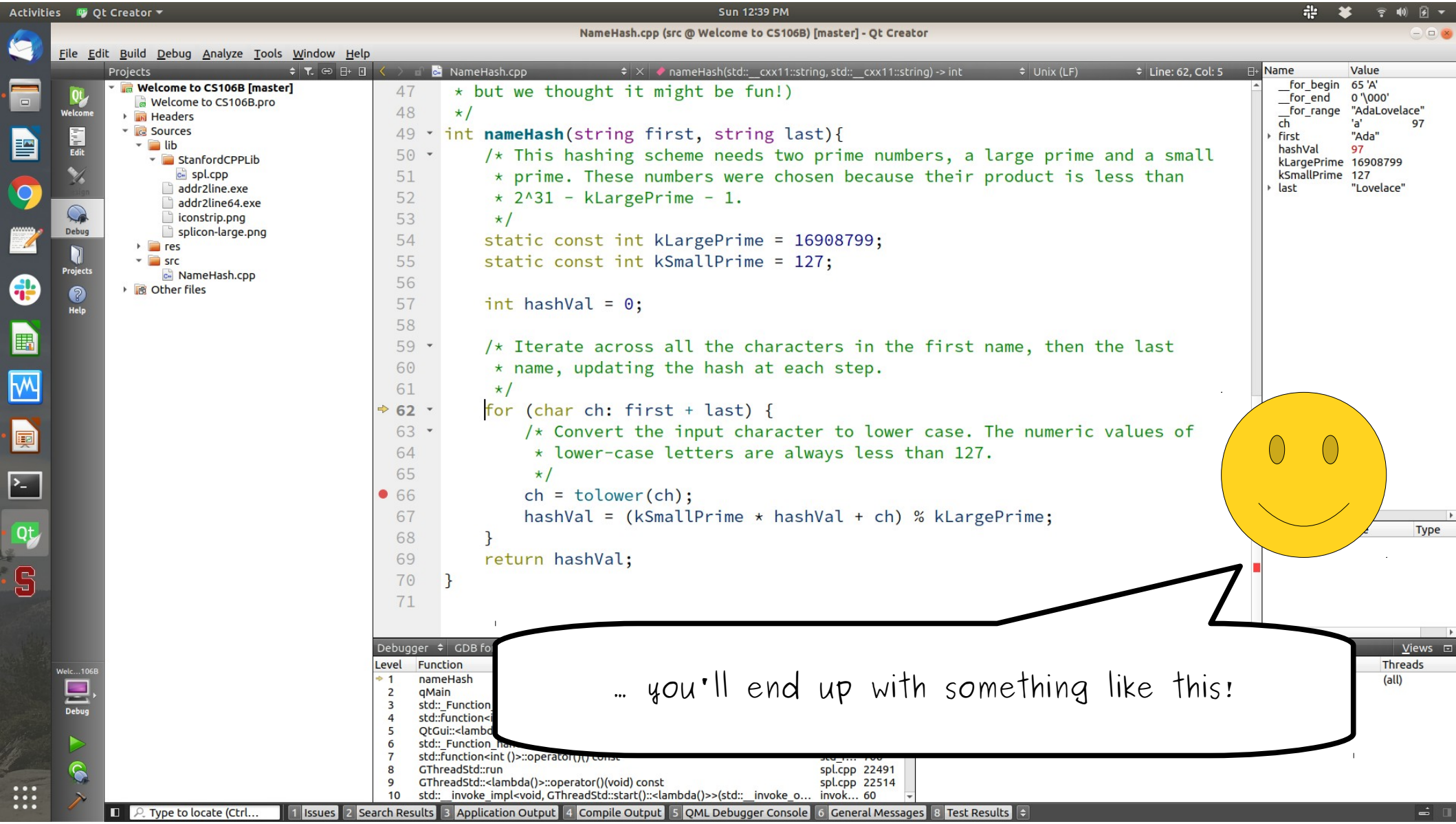
Not gonna lie, this is a pretty dense line of code. It performs some weird sort of mathematical calculation on a bunch of different values.

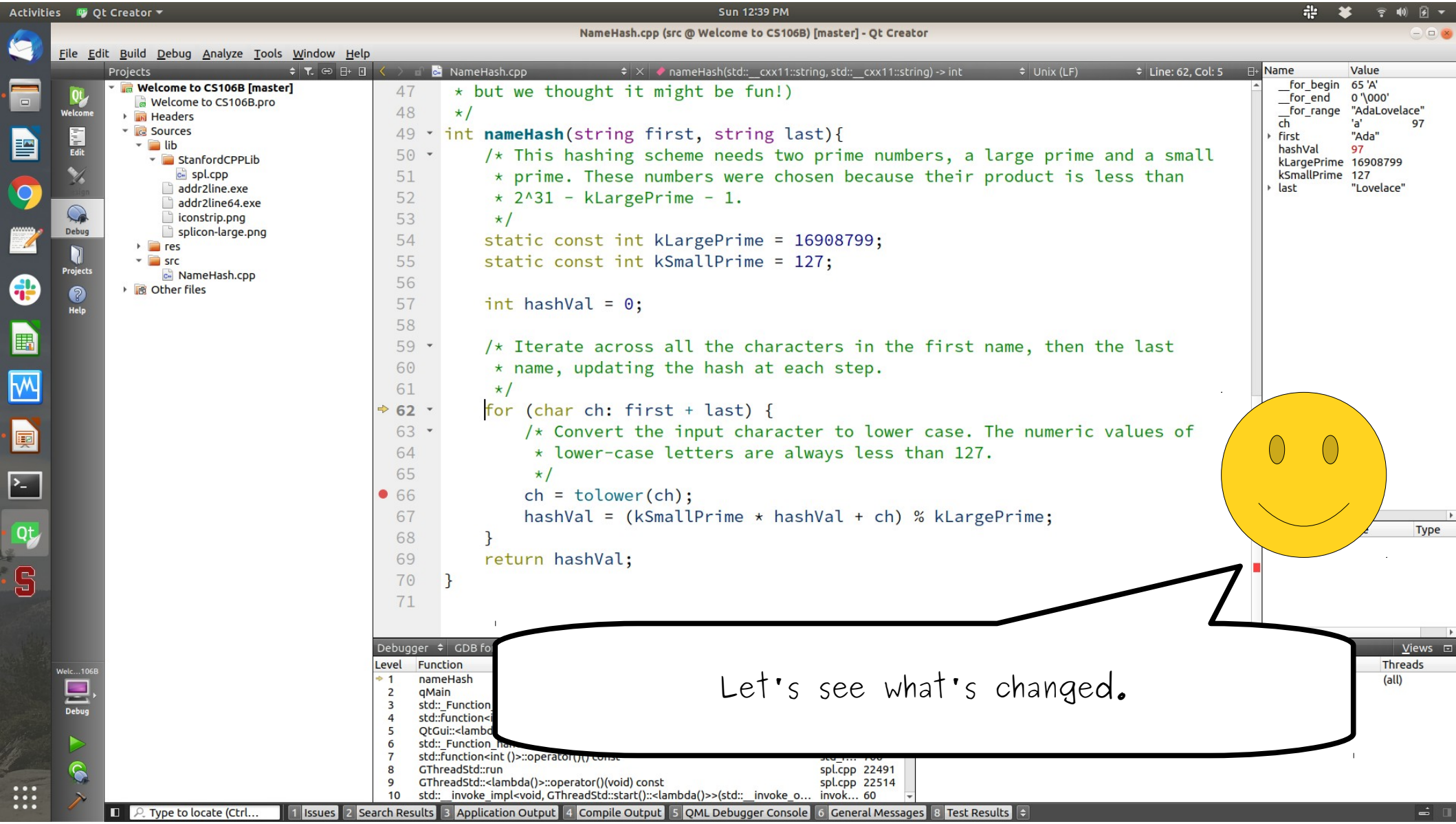
```
55
56
57      int hashVal = 0;
58
59      /* Iterate across all the characters in the first name, then the last
60       * name, updating the hash at each step.
61       */
62      for (char ch: first + last) {
63          /* Convert the input character to lower case. The numeric values of
64   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


Activities
Qt Creator
Sun 12:35 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
Fundamentally, though, it's just computing some weird function of some values and stashing it into hashVal.
int hashVal = 0;
/* Iterate across all the characters in the first name, then the last
* name, updating the hash at each step.
*/
for (char ch: first + last) {
/* Convert the input character to lower case. The numeric values of
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Debugger GDB for "Welcome to CS106B" Threads: #7 Welcome_to_CS10 Stopped: "end-stepping-range".
Level Function File Line
1 nameHash
2 qMain
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&)
4 std::function<int ()>::operator()() const
5 QtGui::<lambda()>::operator()(void) const
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn...
7 std::function<int ()>::operator()() const
8 GThreadStd::run
9 GThreadStd::<lambda()>::operator()(void) const
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o...
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results


Activities
Qt Creator
Sun 12:35 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
Let's go run that line of code and see what happens!
int hashVal = 0;
/* Iterate across all the characters in the first name, then the last
* name, updating the hash at each step.
*/
for (char ch: first + last) {
/* Convert the input character to lower case. The numeric values of
* lower-case letters are always less than 127.
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}
return hashVal;
}
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam... 67
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

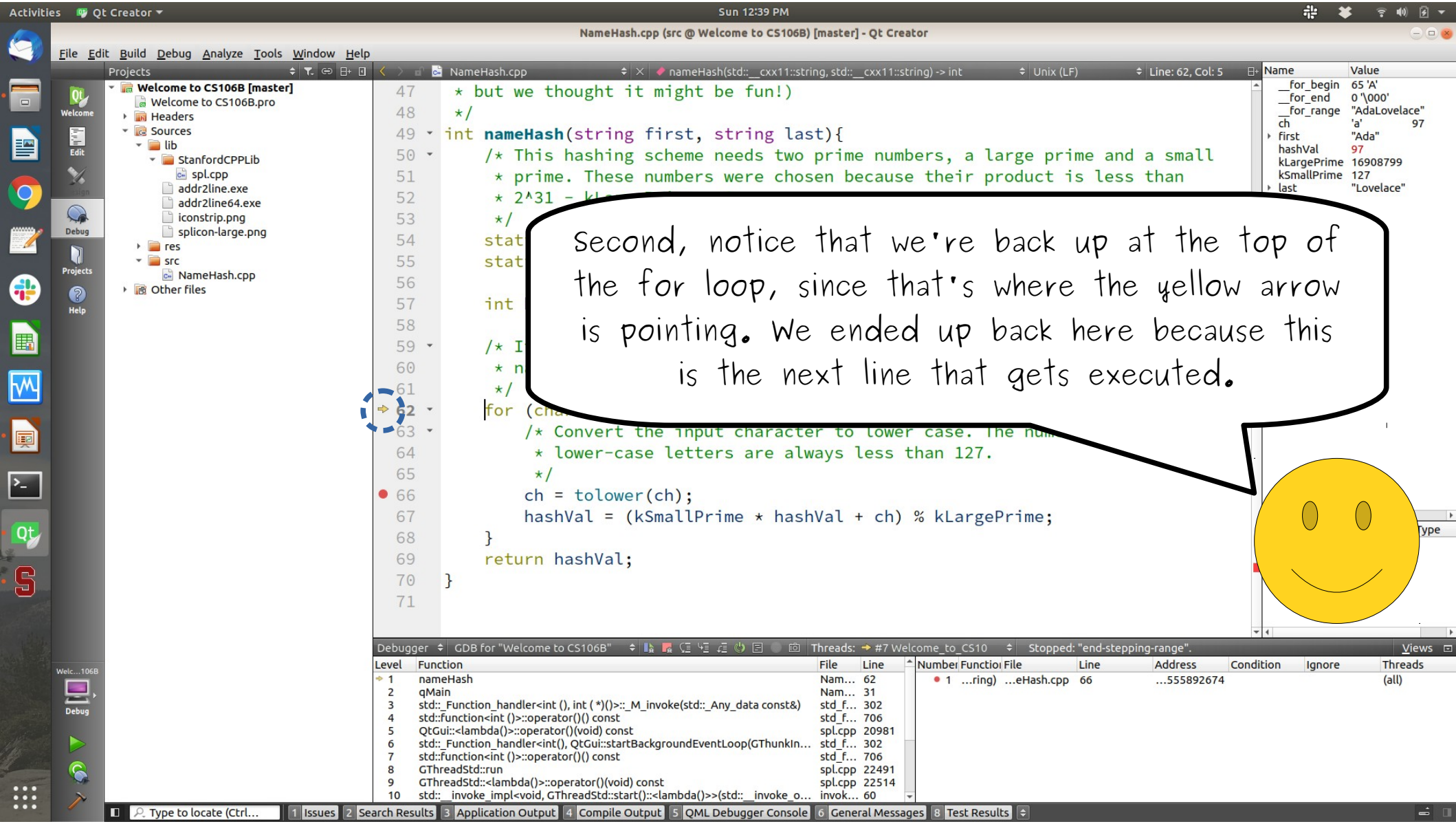

Activities
Qt Creator
Sun 12:35 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 67, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then t
60 * name, updating the hash at each step.
61 */
70
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 0
kLargePrime 16908799
Name Value Type
Hover over the "Step Over" button, confirm that the button you're clicking really is "Step Over," and click it! When you do…
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam… 67
2 qMain Nam… 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f… 302
4 std::function<int ()>::operator()() const std_f… 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn… std_f… 302
7 std::function<int ()>::operator()() const std_f… 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o… invok… 60
Number Functio File Line Address Condition Ignore Threads
1 …ring) …eHash.cpp 66 …555892674 (all)
Type to locate (Ctrl…
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

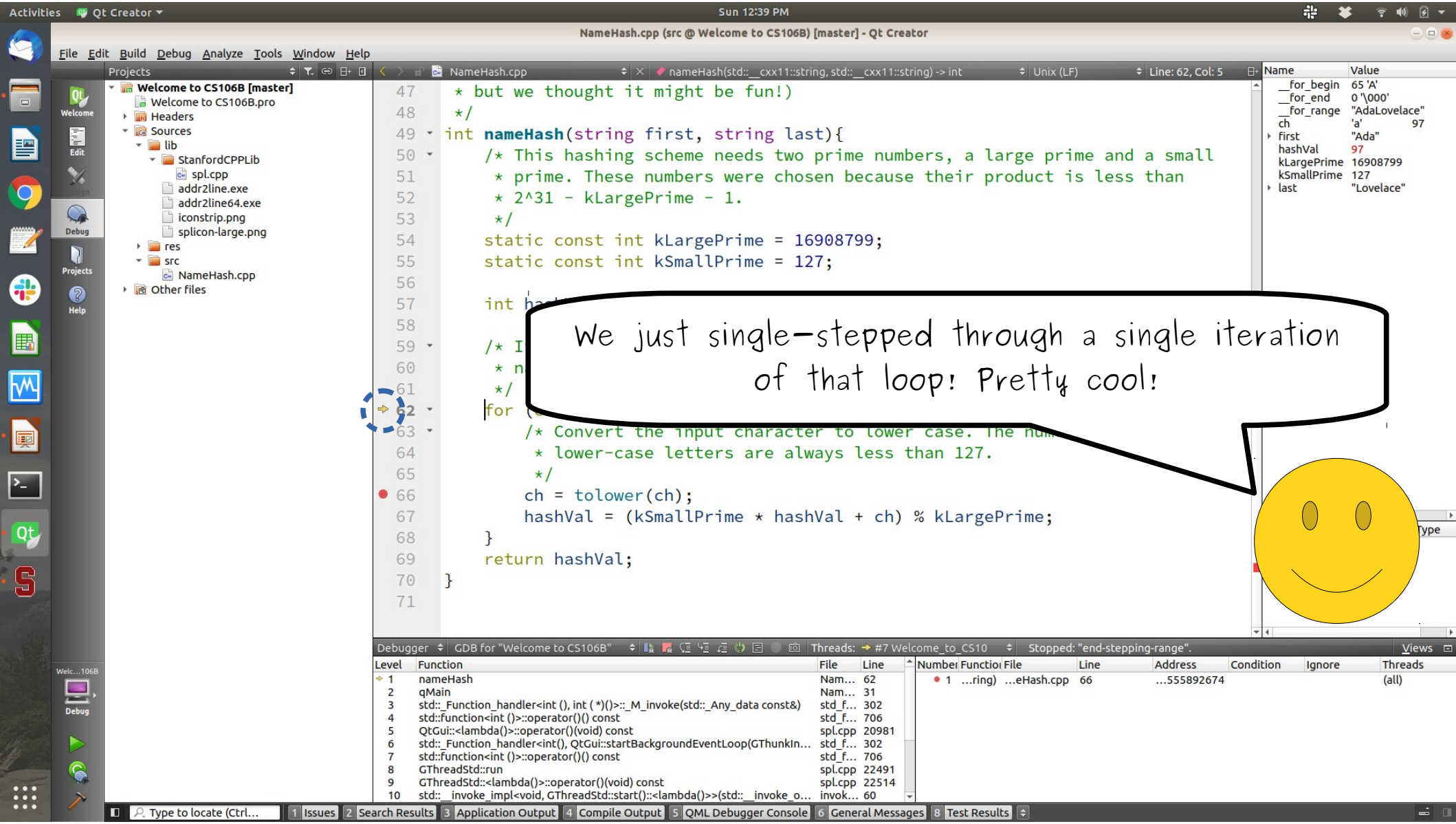

Activities
Qt Creator
Sun 12:39 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 97
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Type
Debugger
Views
Level Function
1 nameHash
2 qMain
3 std::_Function_
4 std::function<
5 QtGui::<lambd
6 std::_Function_ha
7 std::function<int ()>::operator()() const
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Threads
(all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results
... you'll end up with something like this!

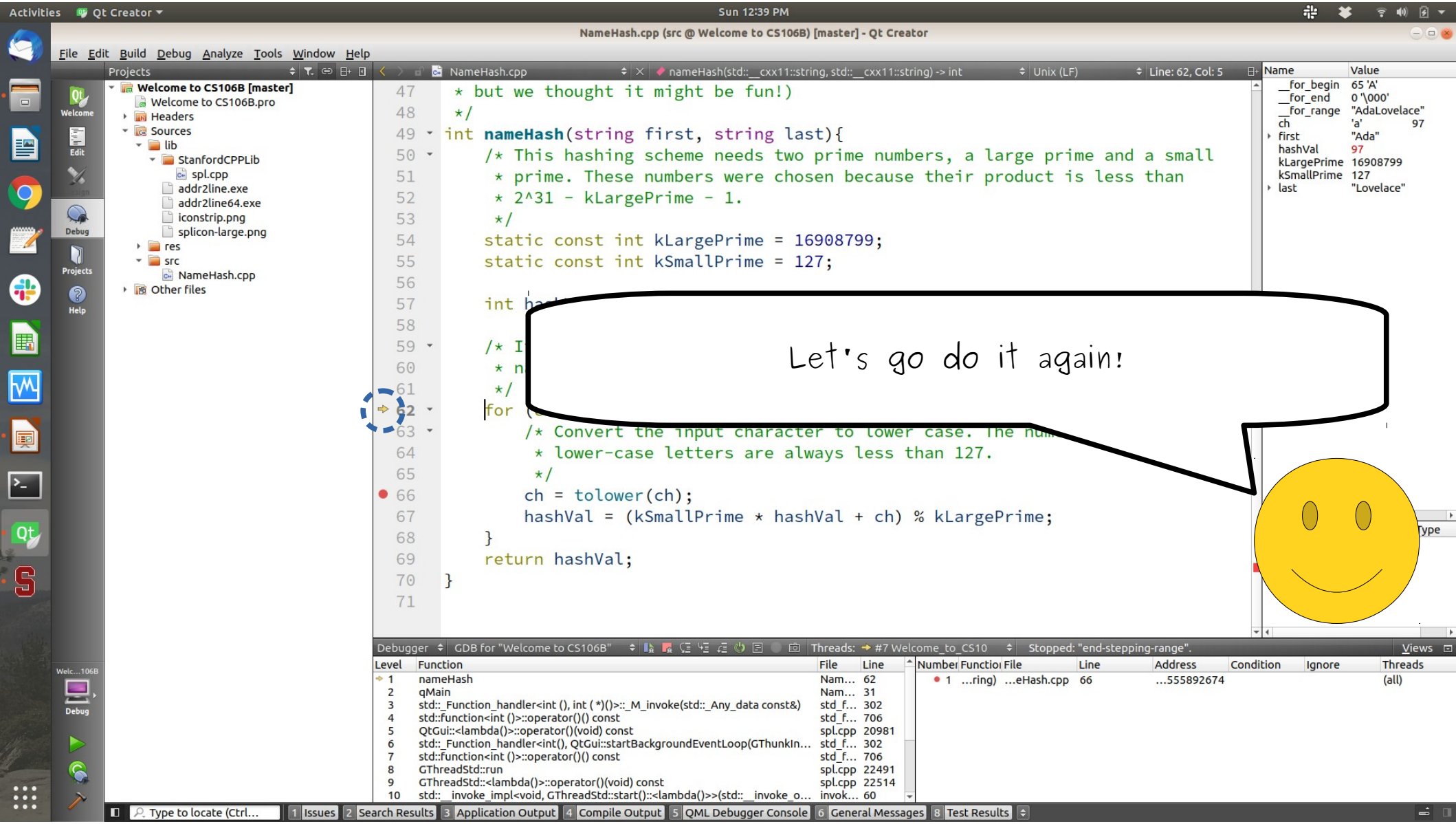

Activities
Qt Creator
Sun 12:39 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then the last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The numeric values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 97
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Type
Debugger
Views
Level Function
1 nameHash
2 qMain
3 std::_Function_
4 std::function<
5 QtGui::<lambd
6 std::_Function_ha
7 std::function<int ()>::operator()() const
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Threads
(all)
Let's see what's changed.
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

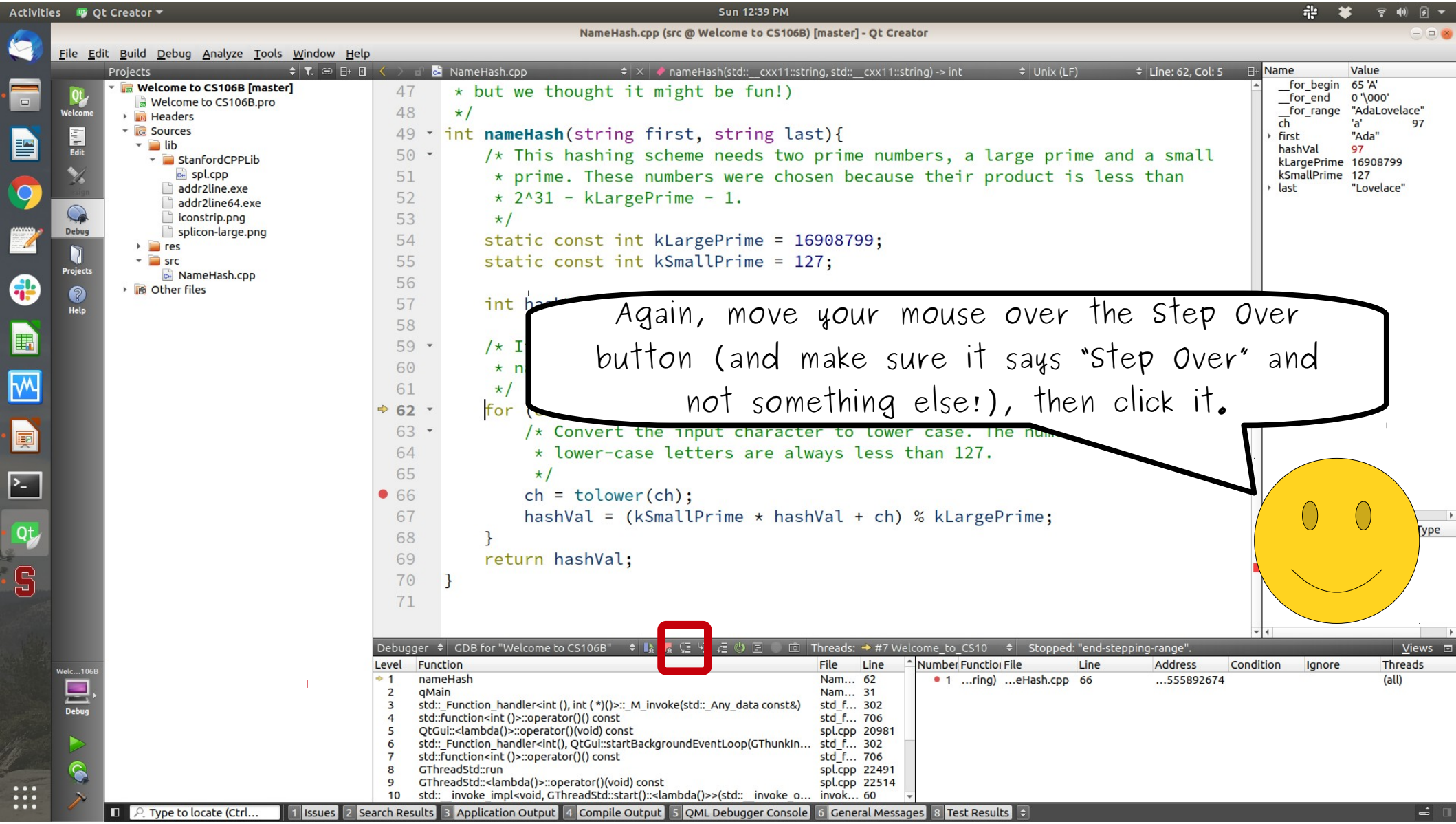
First, notice that the value stored in **hashVal** changed to 97. We know that it changed because the value is in red, and we know that nothing else changed because nothing else is in red!

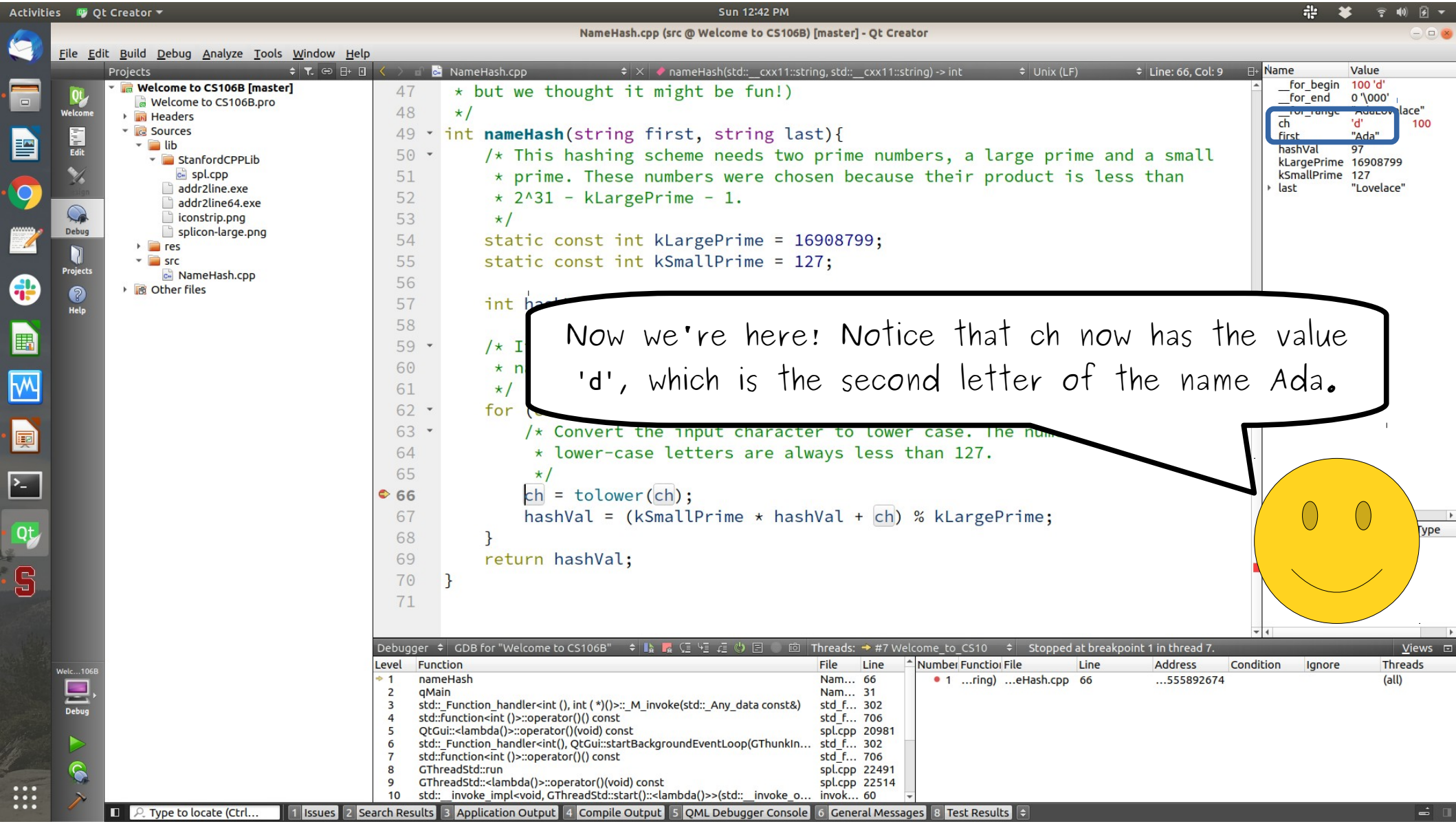
```
52
53
54      static const int kLargePrime = 16908799;
55      static const int kSmallPrime = 127;
56
57      int hashVal = 0;
58
59      /* Iterate across all the characters in the first name, then the last
60       * name, updating the hash at each step.
61       */
62      for (char ch: first + last) {
63          /* Convert the input character to lower case. The numeric values of
64           * lower-case letters are always less than 127.
65           */
66          ch = tolower(ch);
67          hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68      }
69      return hashVal;
70  }
71
```

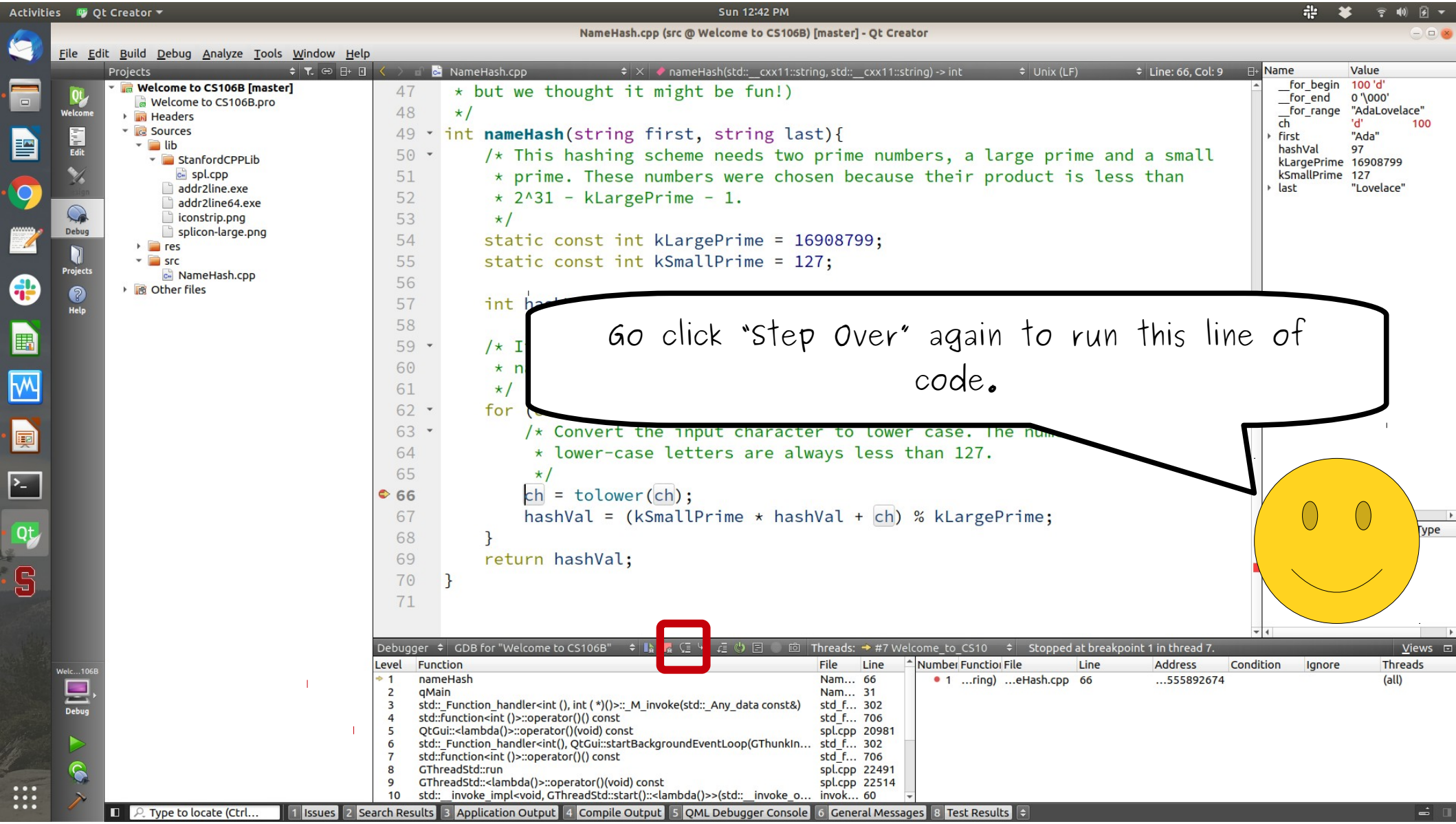

Activities
Qt Creator
Sun 12:39 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kl
53 */
54 stat
55 stat
56
57 int
58
59 /* I
60 * n
61 */
62 for (cha
63 /* Convert the input character to lower case. The nu
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 97
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
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.
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam... 62
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

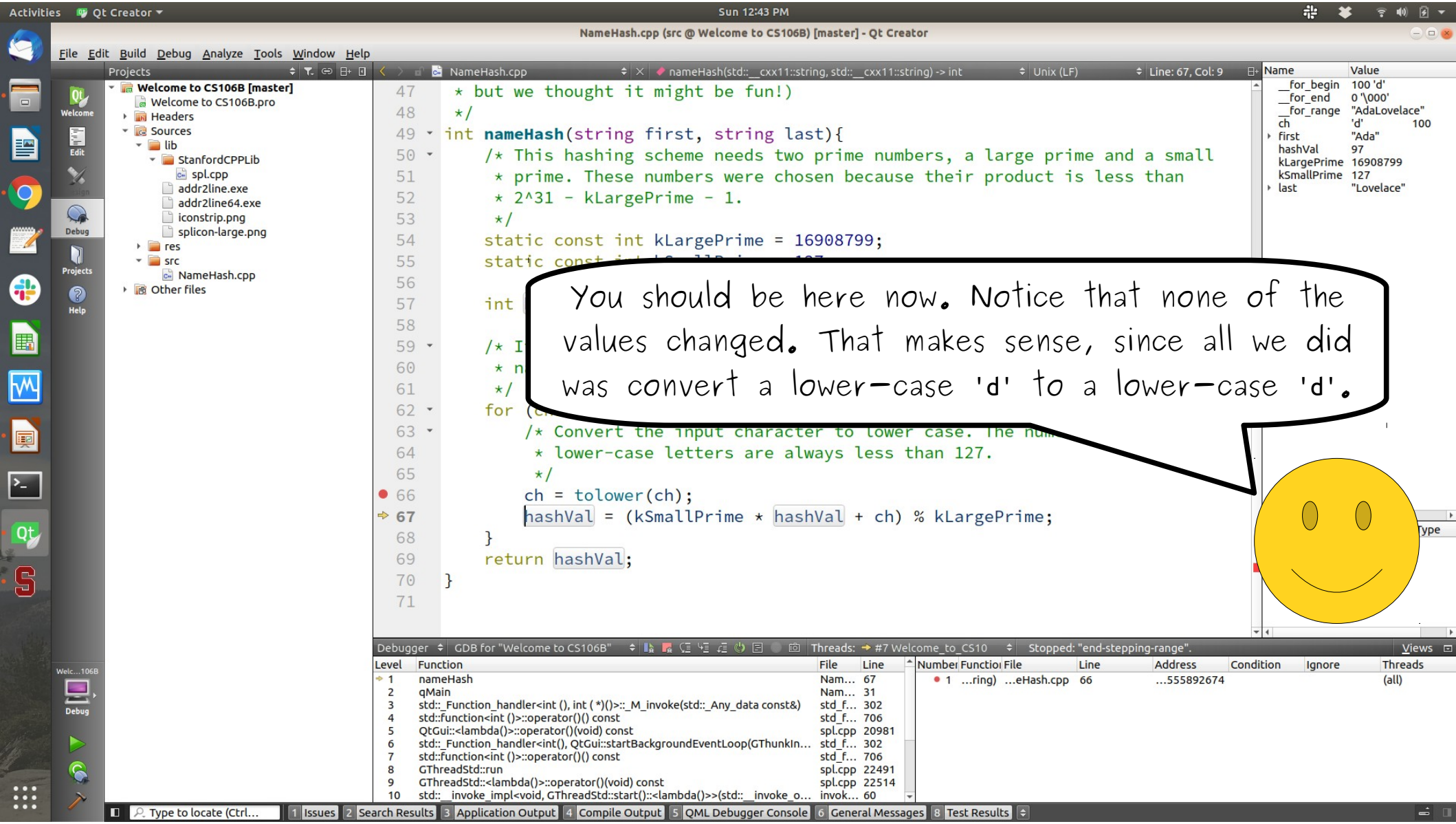

Activities
Qt Creator
Sun 12:39 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int ha
58
59 /* I
60 * n
61 */
62 for (
63 /* Convert the input character to lower case. The nu
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 97
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
We just single-stepped through a single iteration of that loop! Pretty cool!
Debugger GDB for "Welcome to CS106B" Threads: #7 Welcome_to_CS10 Stopped: "end-stepping-range". Views
Level Function File Line
1 nameHash Nam... 62
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

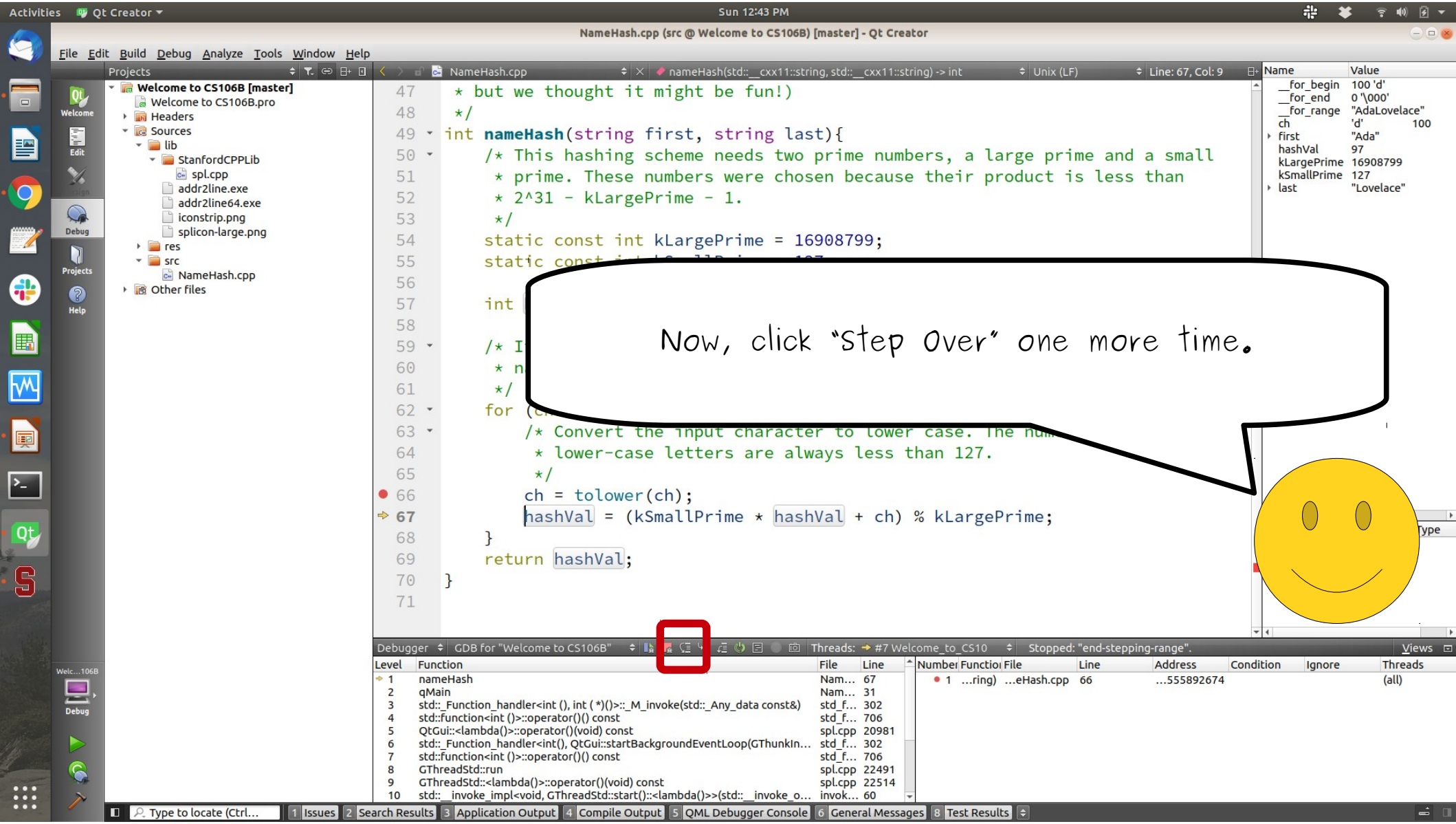

Activities
Qt Creator
Sun 12:39 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int ha
58
59 /* I
60 * n
61 */
62 for (
63 /* Convert the input character to lower case. The nu
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Let's go do it again!
Name
Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 97
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam... 62
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

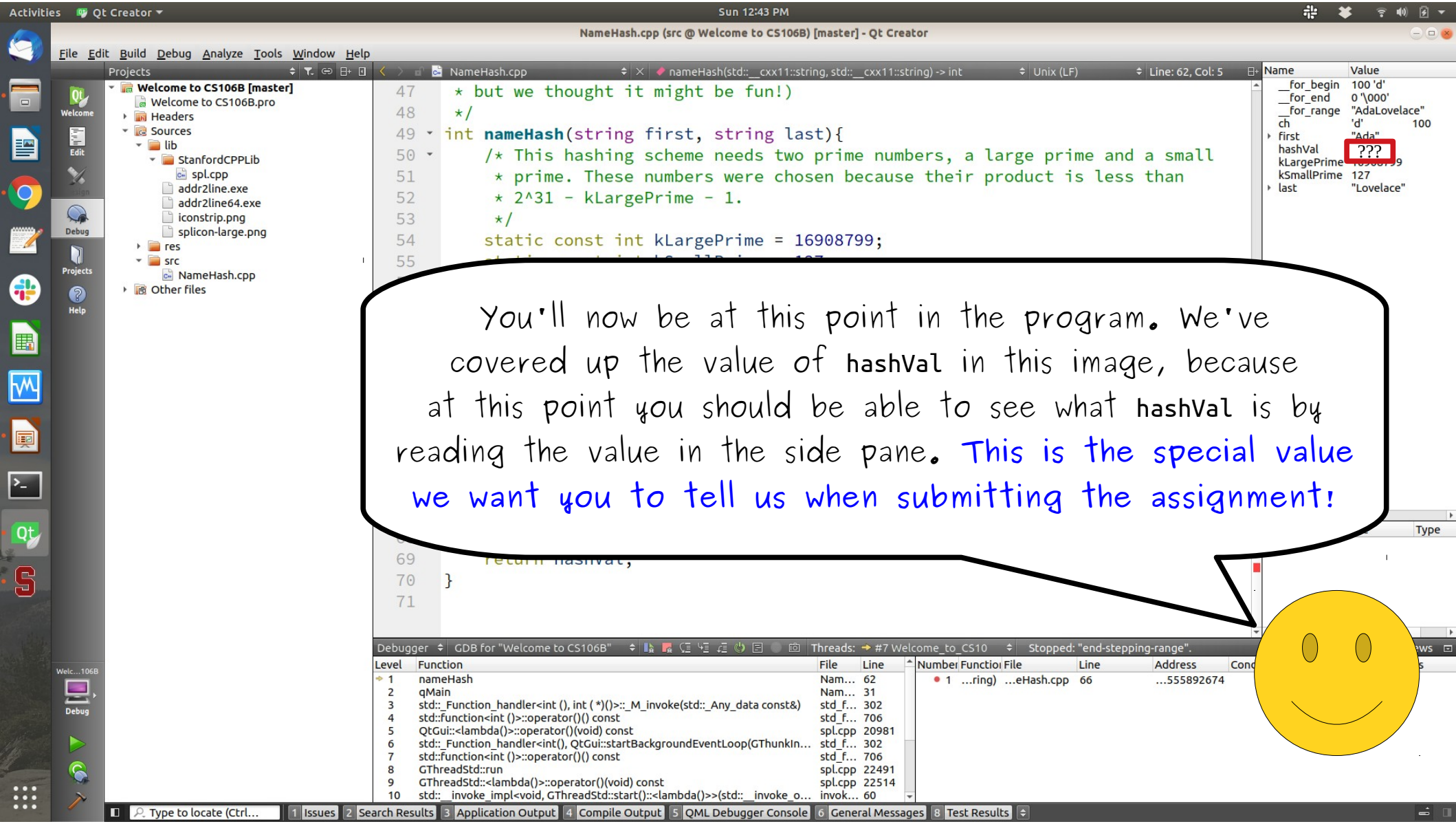

Activities
Qt Creator
Sun 12:39 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int ha
58
59 /* I
60 * n
61 */
62 for (
63 /* Convert the input character to lower case. The nu
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name
Value
__for_begin 65 'A'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'a' 97
first "Ada"
hashVal 97
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Again, move your mouse over the Step Over button (and make sure it says "Step Over" and not something else!), then click it.
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam... 62
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

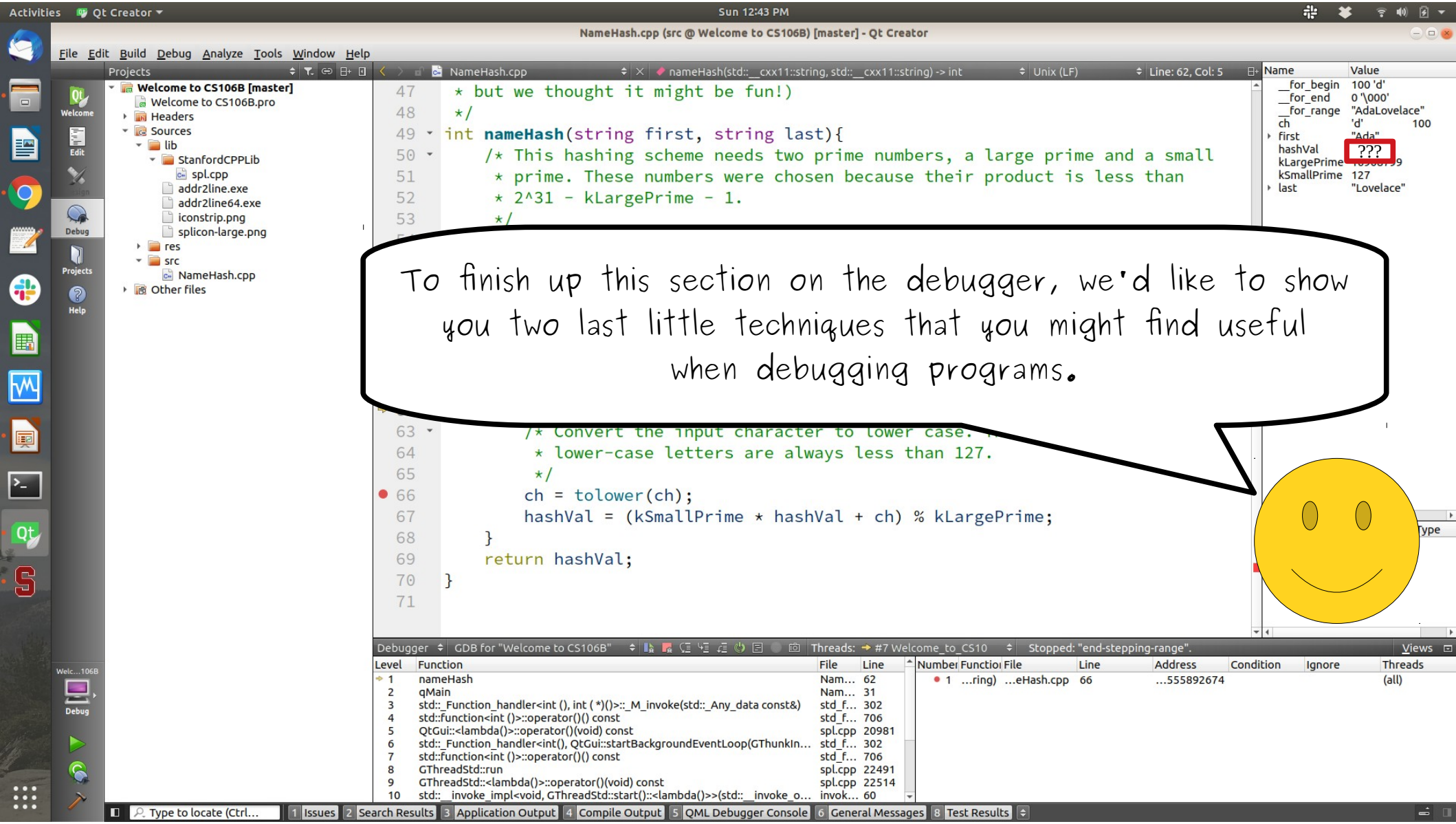

Activities
Qt Creator
Sun 12:42 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int ha
58
59 /* I
60 * n
61 */
62 for (
63 /* Convert the input character to lower case. The nu
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 100 'd'
__for_end 0 '\000'
ch 'd' 100
first "Ada"
hashVal 97
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Now we're here! Notice that ch now has the value 'd', which is the second letter of the name Ada.
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

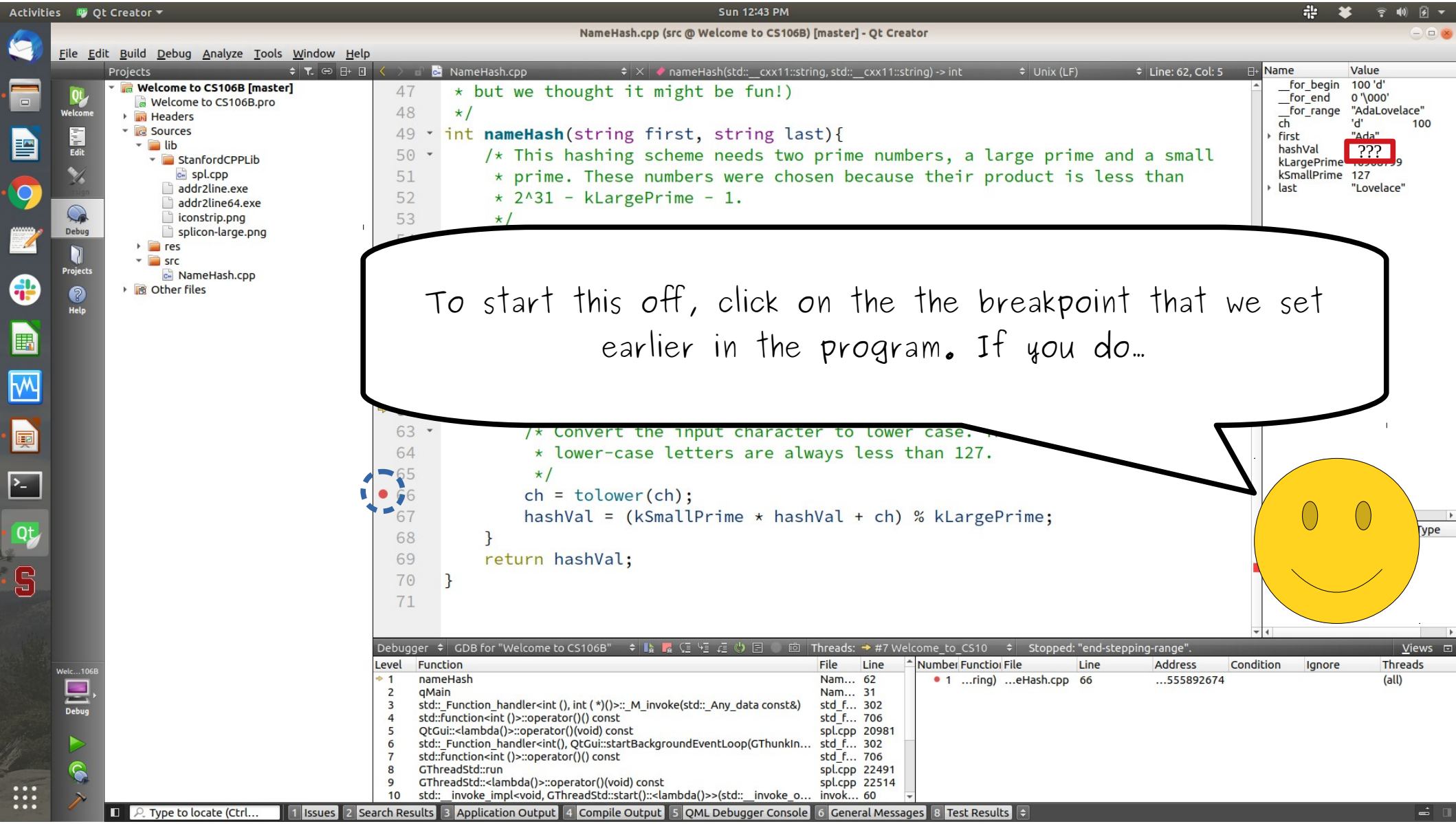

Activities
Qt Creator
Sun 12:42 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 66, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int ha
58
59 /* I
60 * n
61 */
62 for (
63 /* Convert the input character to lower case. The nu
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name
Value
__for_begin 100 'd'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'd' 100
first "Ada"
hashVal 97
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Go click "Step Over" again to run this line of code.
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped at breakpoint 1 in thread 7.
Views
Level Function File Line
1 nameHash Nam... 66
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

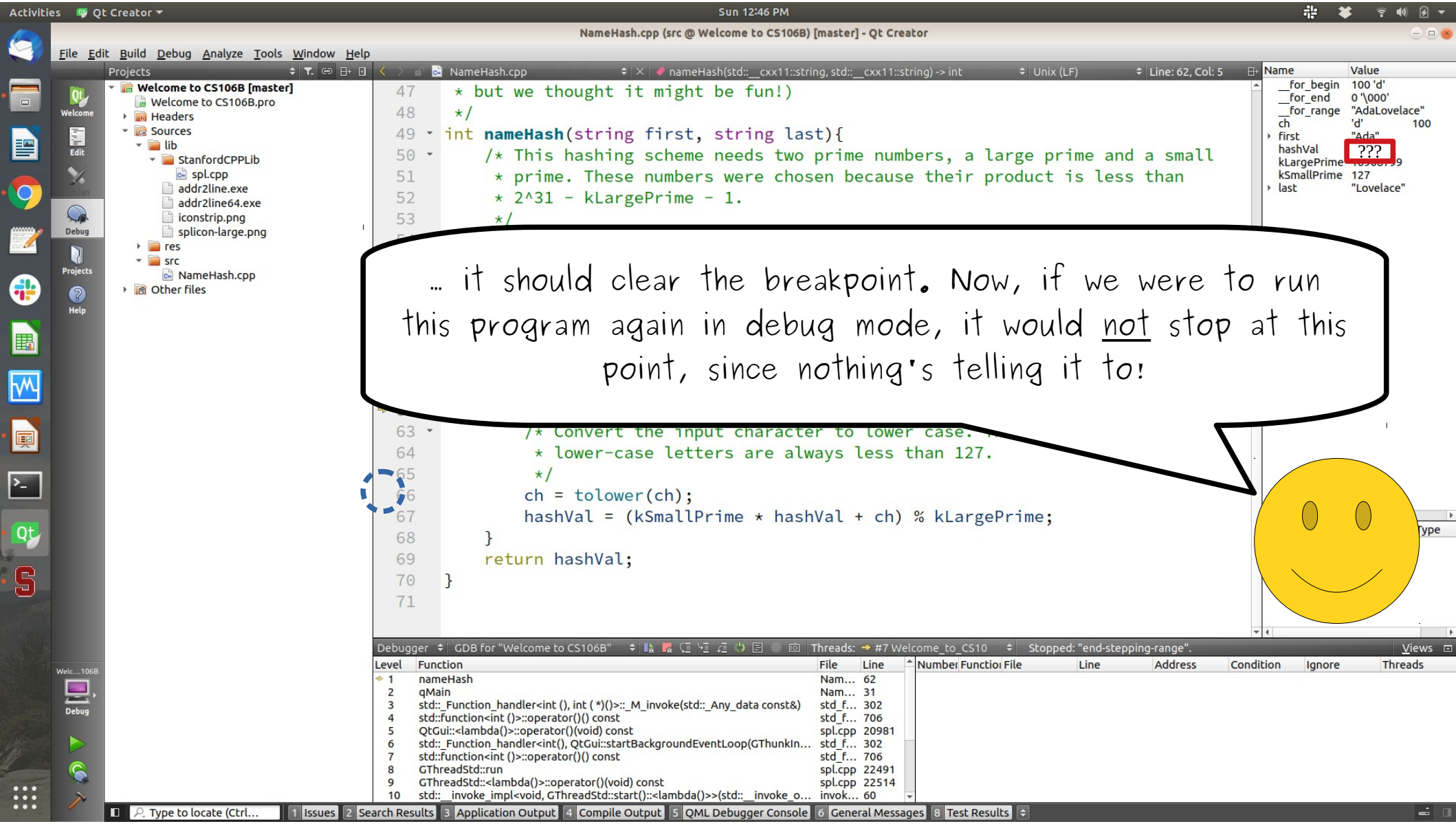

Activities
Qt Creator
Sun 12:43 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 67, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55
56
57
58
59
60
61
62
63
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 100 'd'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'd' 100
first "Ada"
hashVal 97
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
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'.
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam... 67
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

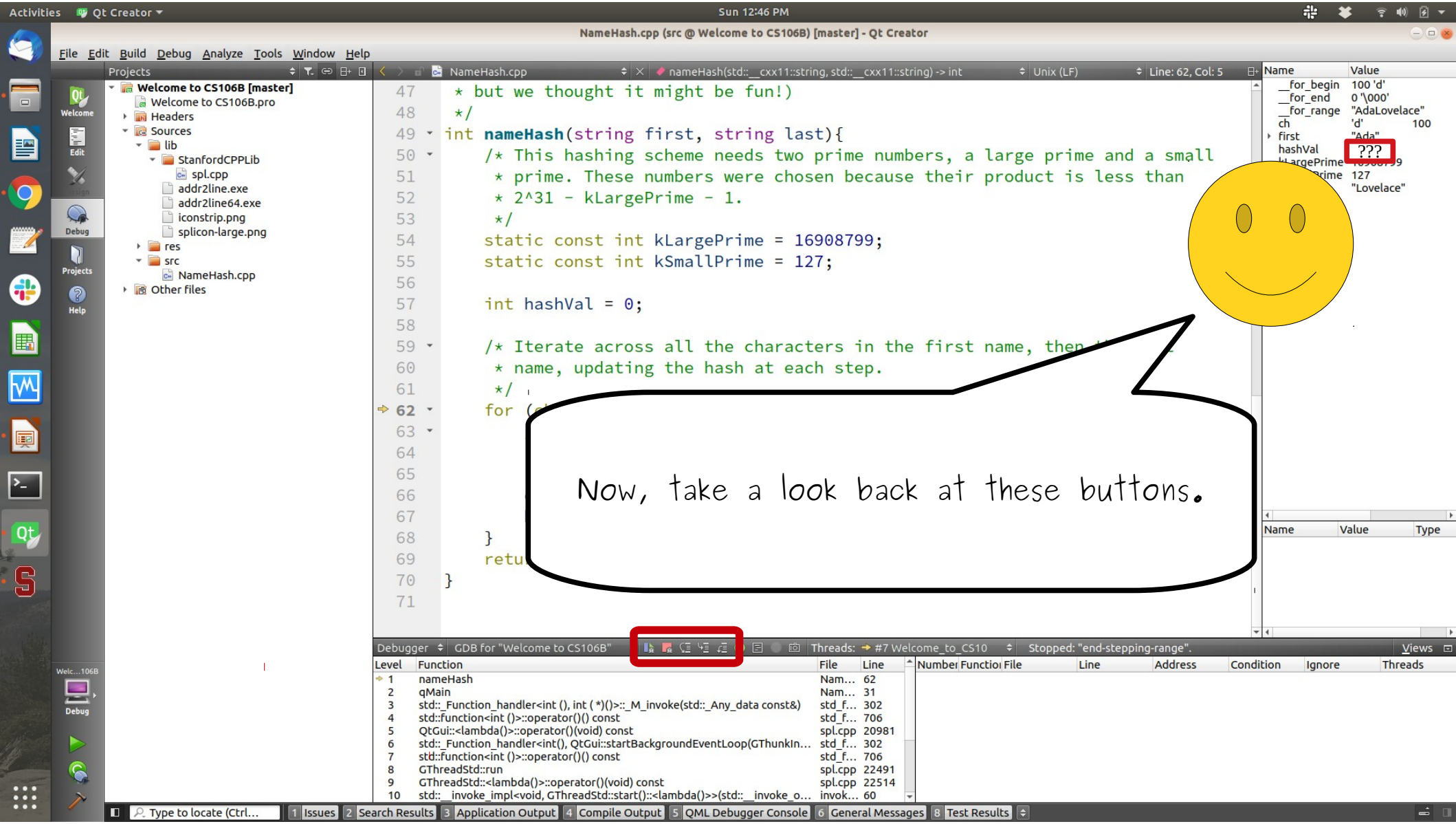

Activities
Qt Creator
Sun 12:43 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 67, Col: 9
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55
56
57
58
59
60
61
62
63
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name Value
__for_begin 100 'd'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'd' 100
first "Ada"
hashVal 97
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
Now, click "Step Over" one more time.
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam... 67
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

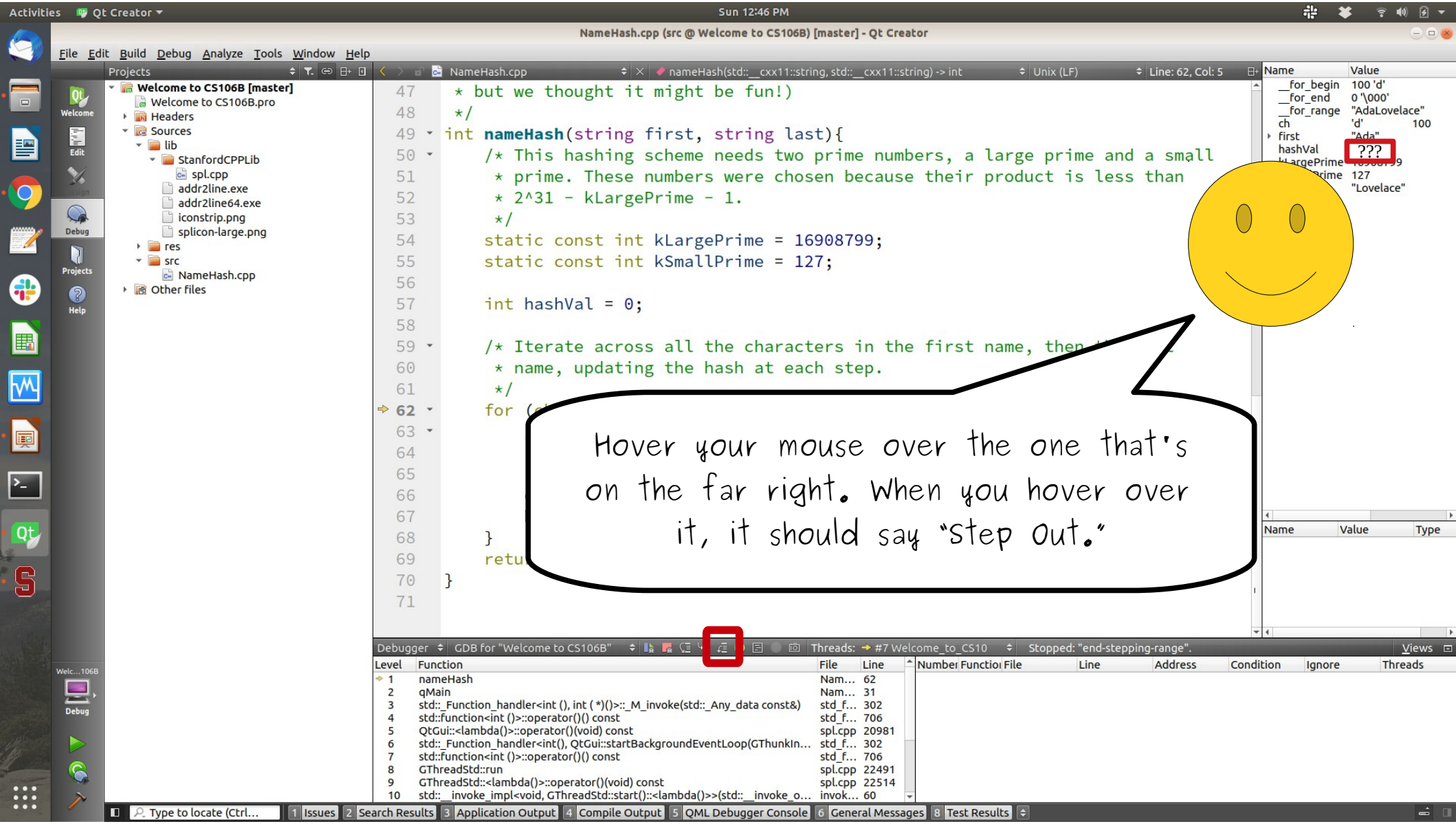

Activities
Qt Creator
Sun 12:43 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
69 return hashVal;
70 }
71
Name
Value
__for_begin 100 'd'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'd' 100
first "Ada"
hashVal ???
kLargePrime
kSmallPrime 127
last "Lovelace"
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!
Type
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped: "end-stepping-range".
Level Function File Line
1 nameHash Nam... 62
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address
1 ...ring) ...eHash.cpp 66 ...555892674
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results
Welc...106B
Debug

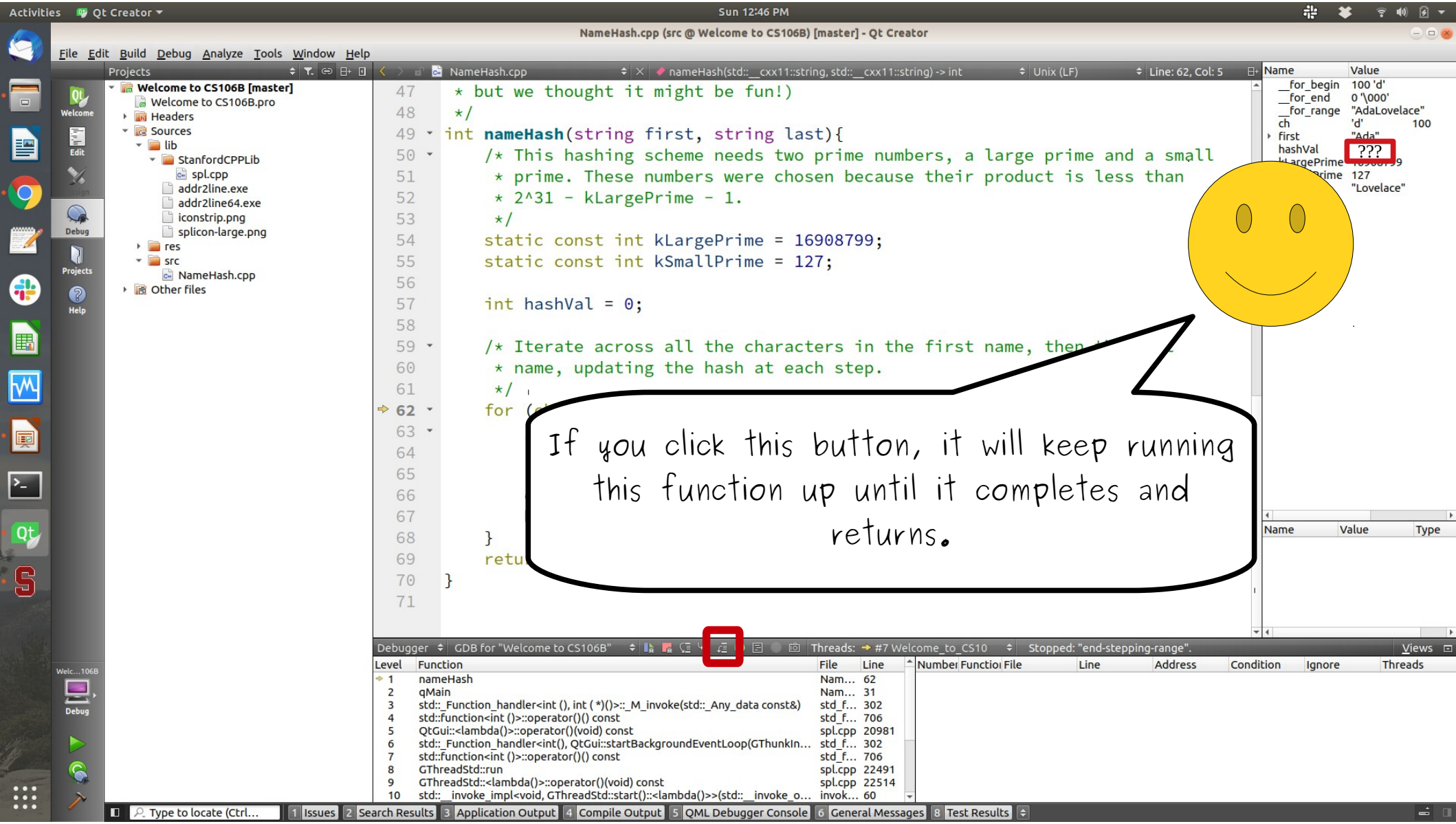

Activities
Qt Creator
Sun 12:43 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
63 /* Convert the input character to lower case.
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name
Value
__for_begin
100 'd'
__for_end
0 '\000'
__for_range
"AdaLovelace"
ch
'd'
100
first
"Ada"
hashVal
???
kLargePrime
kSmallPrime
127
last
"Lovelace"
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.
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level
Function
File
Line
1 nameHash Nam... 62
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number
Function
File
Line
Address
Condition
Ignore
Threads
1 ...ring) ...eHash.cpp 66 ...555892674 (all)
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

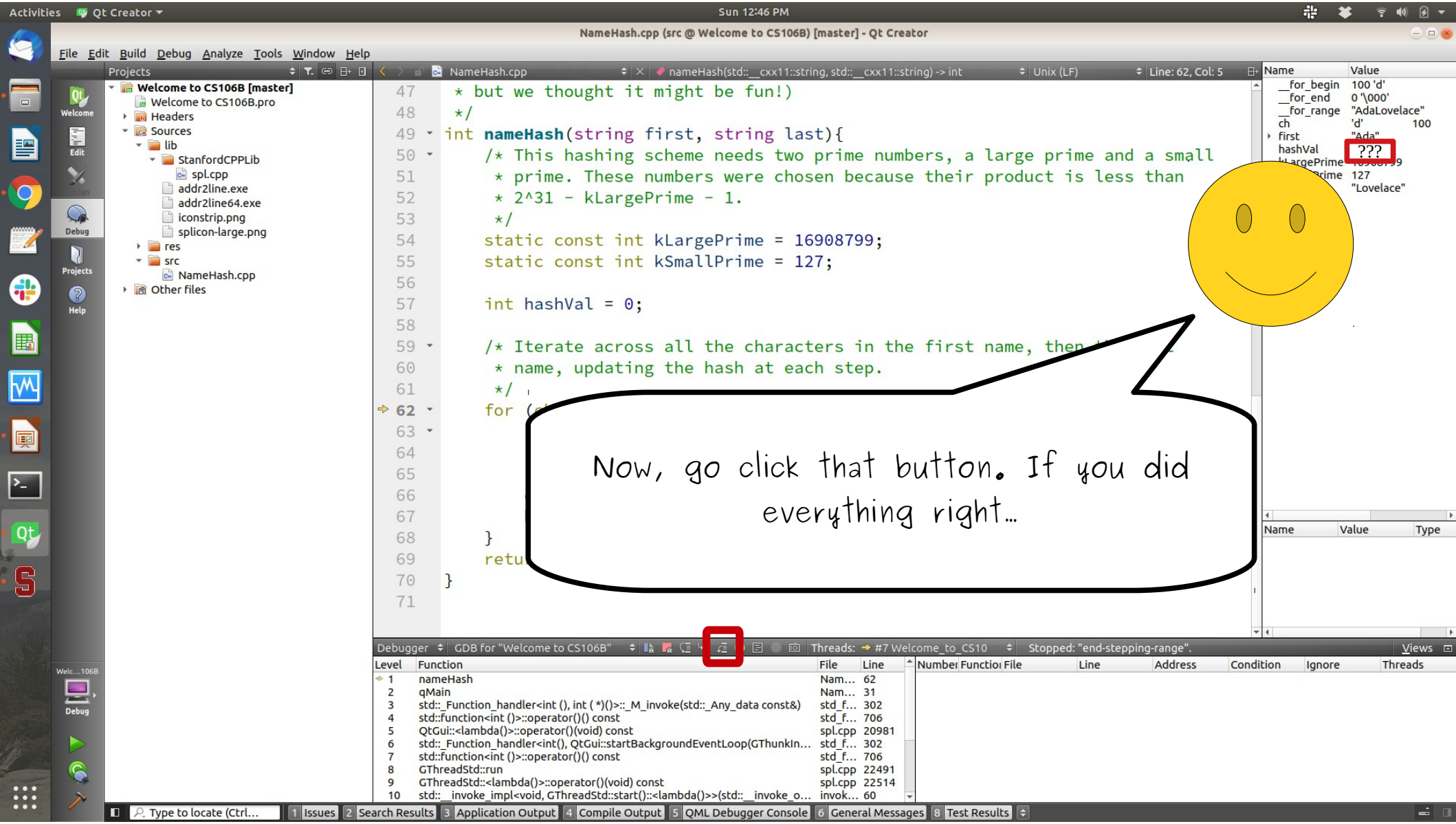

Activities
Qt Creator
Sun 12:43 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
63 /* Convert the input character to lower case.
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name
Value
__for_begin
100 'd'
__for_end
0 '\000'
__for_range
"AdaLovelace"
ch
'd' 100
first
"Ada"
hashVal
???
kLargePrime
kSmallPrime
127
last
"Lovelace"
To start this off, click on the the breakpoint that we set earlier in the program. If you do…
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam… 62
2 qMain Nam… 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f… 302
4 std::function<int ()>::operator()() const std_f… 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn… std_f… 302
7 std::function<int ()>::operator()() const std_f… 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o… invok… 60
Number Functio File Line Address Condition Ignore Threads
1 …ring) …eHash.cpp 66 …555892674 (all)
Type to locate (Ctrl…
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

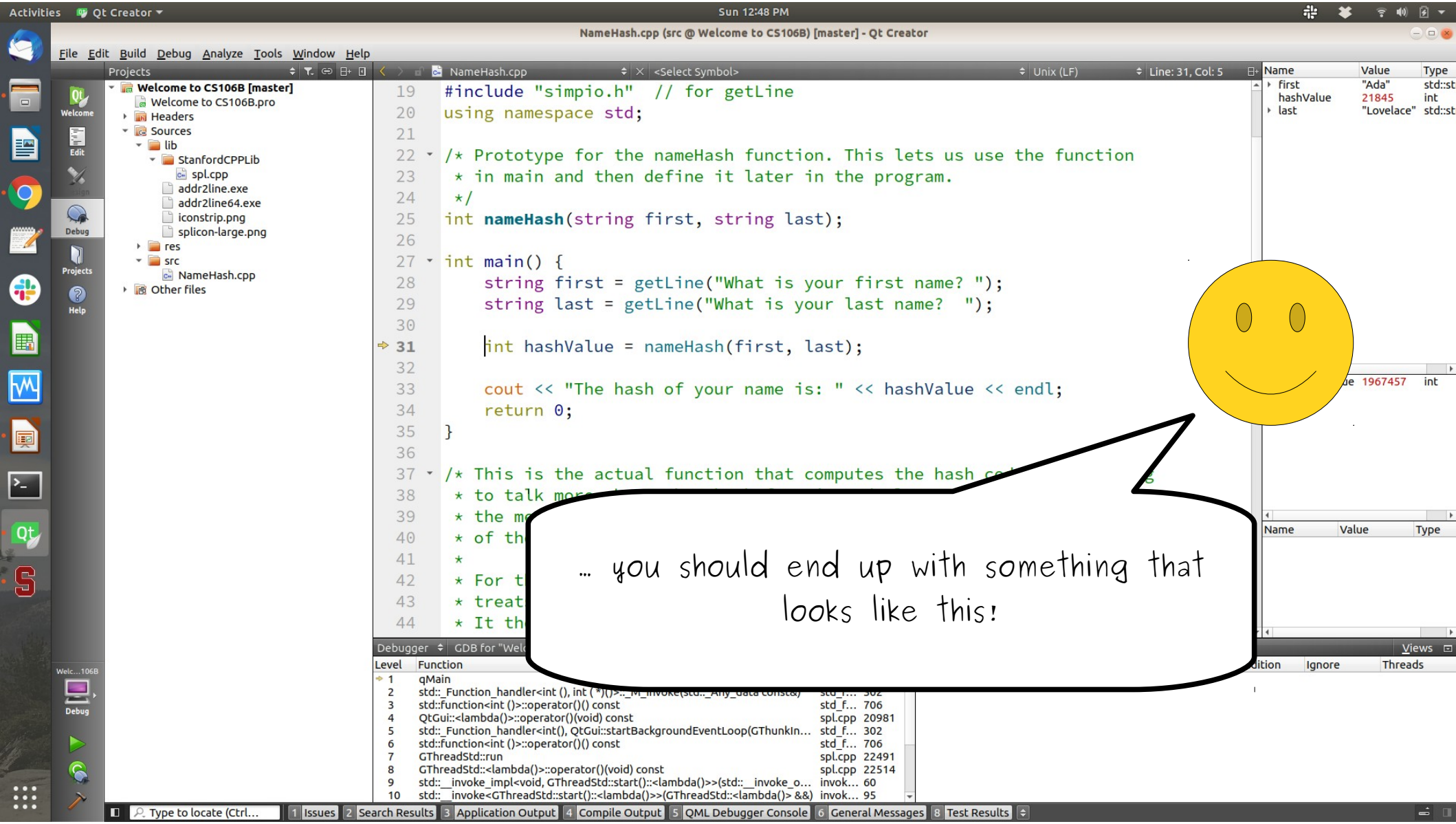

Activities
Qt Creator
Sun 12:46 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
63 /* Convert the input character to lower case.
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
Name
Value
__for_begin
100 'd'
__for_end
0 '\000'
__for_range
"AdaLovelace"
ch
'd'
100
first
"Ada"
hashVal
???
kLargePrime
kSmallPrime
127
last
"Lovelace"
… 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!
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level
Function
File
Line
1 nameHash
2 qMain
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&)
4 std::function<int ()>::operator()() const
5 QtGui::<lambda()>::operator()(void) const
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn…
7 std::function<int ()>::operator()() const
8 GThreadStd::run
9 GThreadStd::<lambda()>::operator()(void) const
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o…
Number
Function
File
Line
Address
Condition
Ignore
Threads
Type to locate (Ctrl…
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

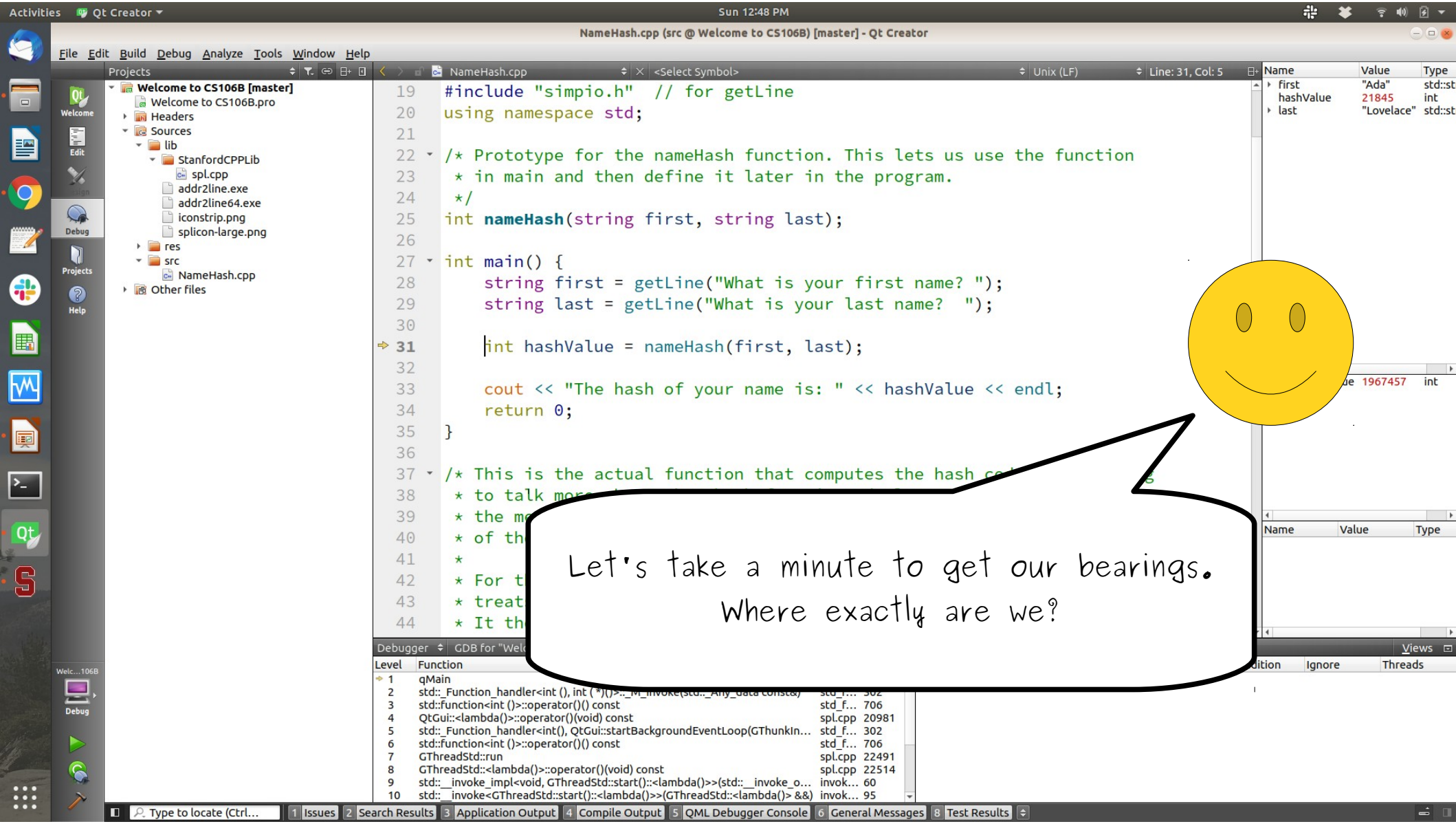

Activities
Qt Creator
Sun 12:46 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name Value
__for_begin 100 'd'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'd' 100
first "Ada"
hashVal ???
kLargePrime
rime 127
"Lovelace"
Name Value Type
Now, take a look back at these buttons.
Debugger GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam... 62
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

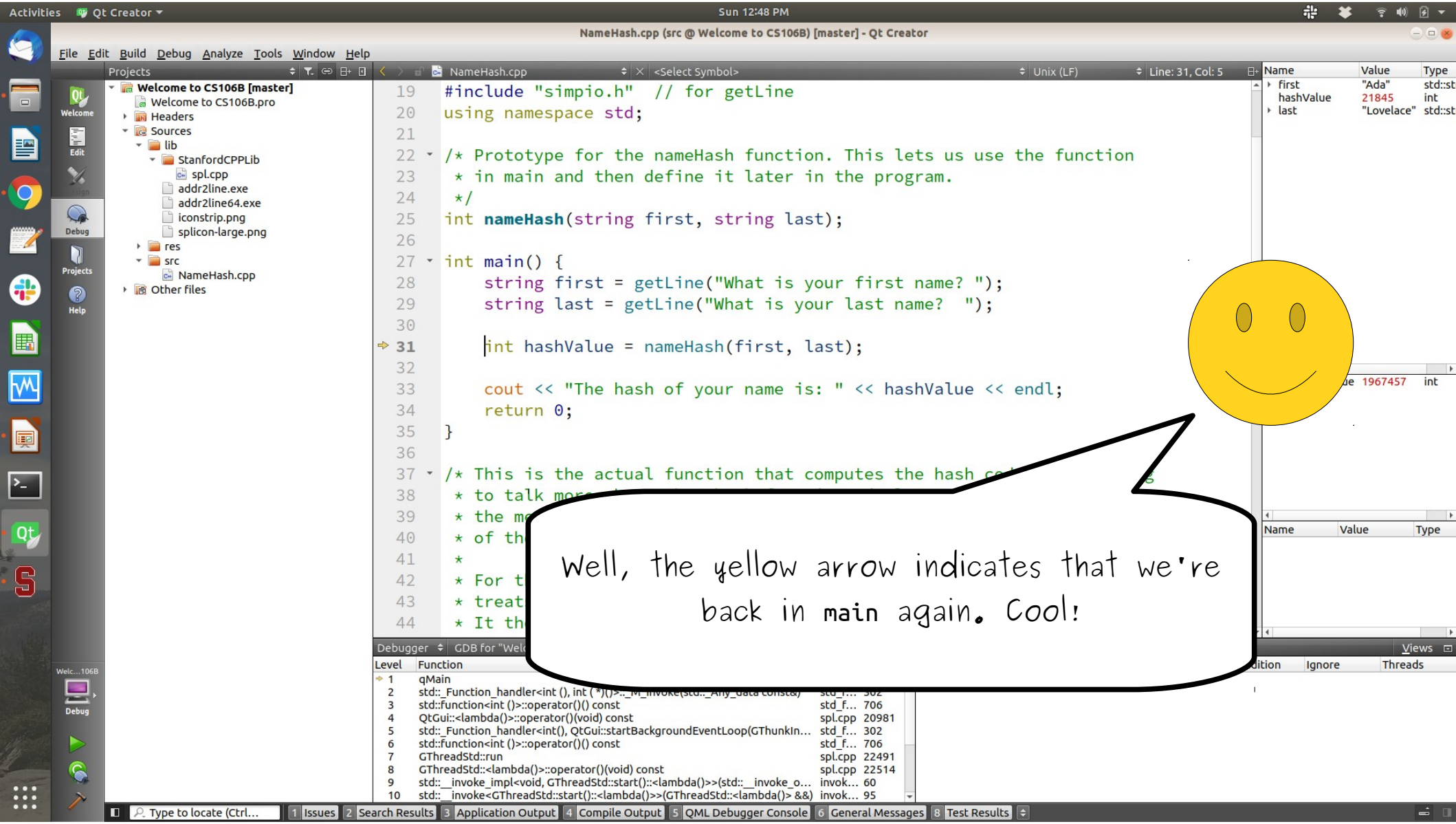

Activities
Qt Creator
Sun 12:46 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name Value
__for_begin 100 'd'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'd' 100
first "Ada"
hashVal ???
kLargePrime
rime 127
"Lovelace"
Name Value Type
Hover your mouse over the one that's on the far right. When you hover over it, it should say "Step Out."
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam... 62
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

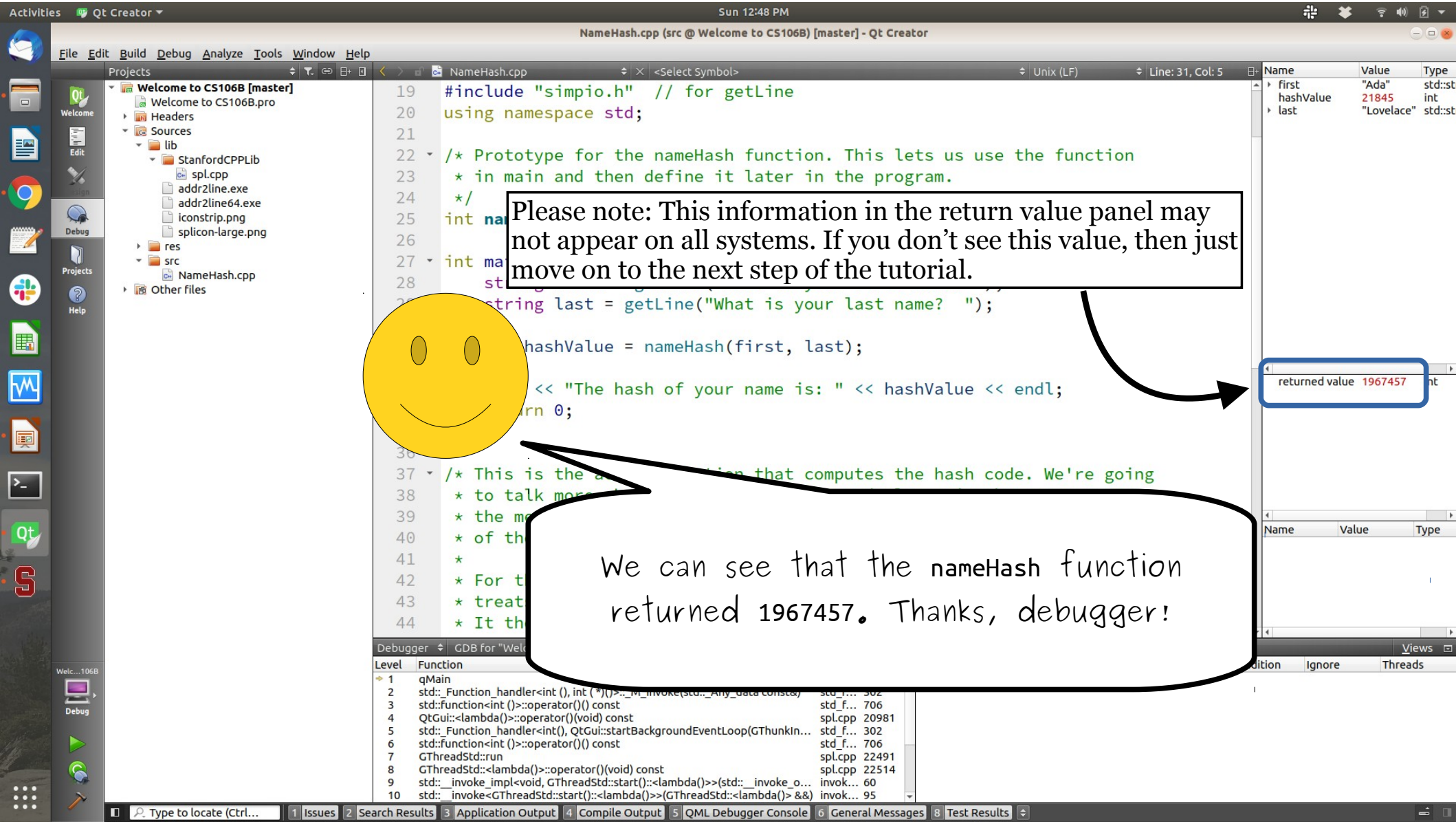

Activities
Qt Creator
Sun 12:46 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name Value
__for_begin 100 'd'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'd' 100
first "Ada"
hashVal ???
kLargePrime
rime 127
"Lovelace"
If you click this button, it will keep running this function up until it completes and returns.
Name Value Type
Debugger
GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam... 62
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Functio File Line Address Condition Ignore Threads
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

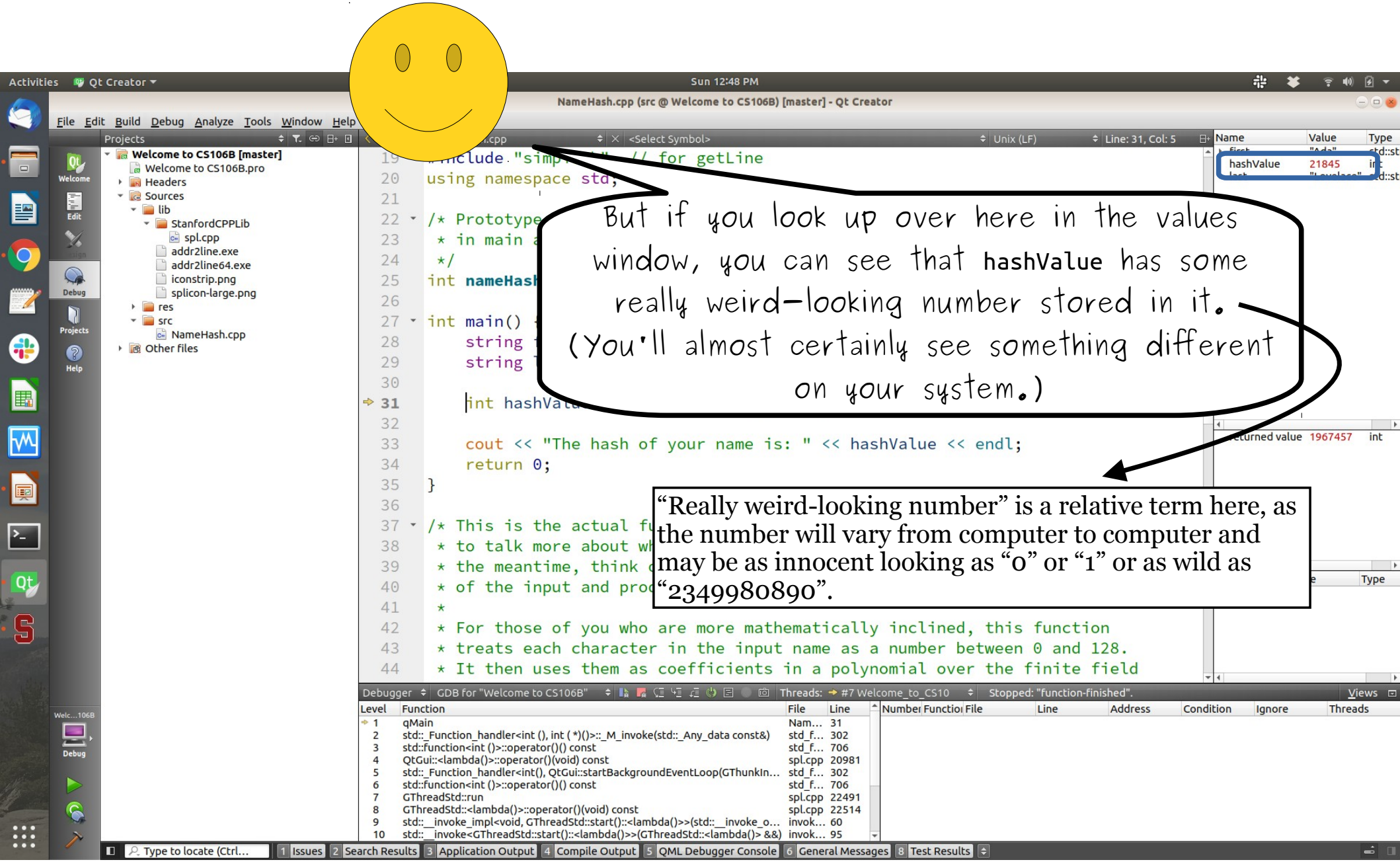

Activities
Qt Creator
Sun 12:46 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
nameHash(std::__cxx11::string, std::__cxx11::string) -> int
Unix (LF)
Line: 62, Col: 5
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
52 * 2^31 - kLargePrime - 1.
53 */
54 static const int kLargePrime = 16908799;
55 static const int kSmallPrime = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first name, then
60 * name, updating the hash at each step.
61 */
62 for (
63
64
65
66
67
68 }
69 retu
70 }
71
Name Value
__for_begin 100 'd'
__for_end 0 '\000'
__for_range "AdaLovelace"
ch 'd' 100
first "Ada"
hashVal ???
127
"Lovelace"
Name Value Type
Now, go click that button. If you did everything right...
Debugger GDB for "Welcome to CS106B"
Threads: #7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 nameHash Nam... 62
2 qMain Nam... 31
3 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f... 302
4 std::function<int ()>::operator()() const std_f... 706
5 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
6 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
7 std::function<int ()>::operator()() const std_f... 706
8 GThreadStd::run spl.cpp 22491
9 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
10 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
Number Function File Line Address Condition Ignore Threads
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

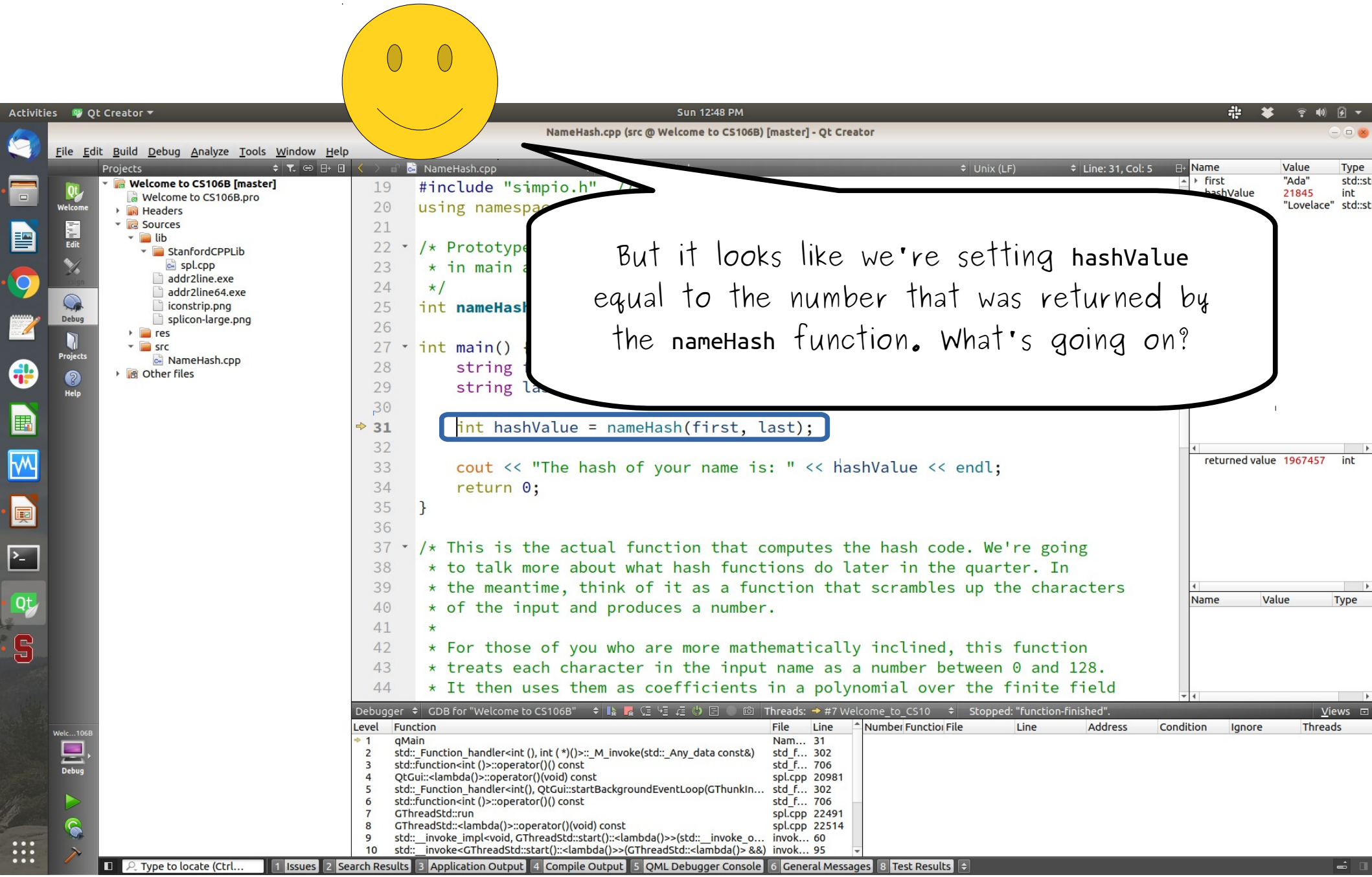

Activities
Qt Creator
Sun 12:48 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 5
#include "simpio.h" // for getLine
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);
cout << "The hash of your name is: " << hashValue << endl;
return 0;
}
/* This is the actual function that computes the hash
* to talk
* the
* of th
*
* For t
* treat
* It th
Name Value Type
first "Ada" std::st
hashValue 21845 int
last "Lovelace" std::st
1967457 int
Debugger
Level Function
1 qMain
2 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&)
3 std::function<int ()>::operator()() const
4 QtGui::<lambda()>::operator()(void) const
5 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn...
6 std::function<int ()>::operator()() const
7 GThreadStd::run
8 GThreadStd::<lambda()>::operator()(void) const
9 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o...
10 std::__invoke<GThreadStd::start()::<lambda()>>(GThreadStd::<lambda()> &&)
Views
Ignore
Threads
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results
… you should end up with something that looks like this!

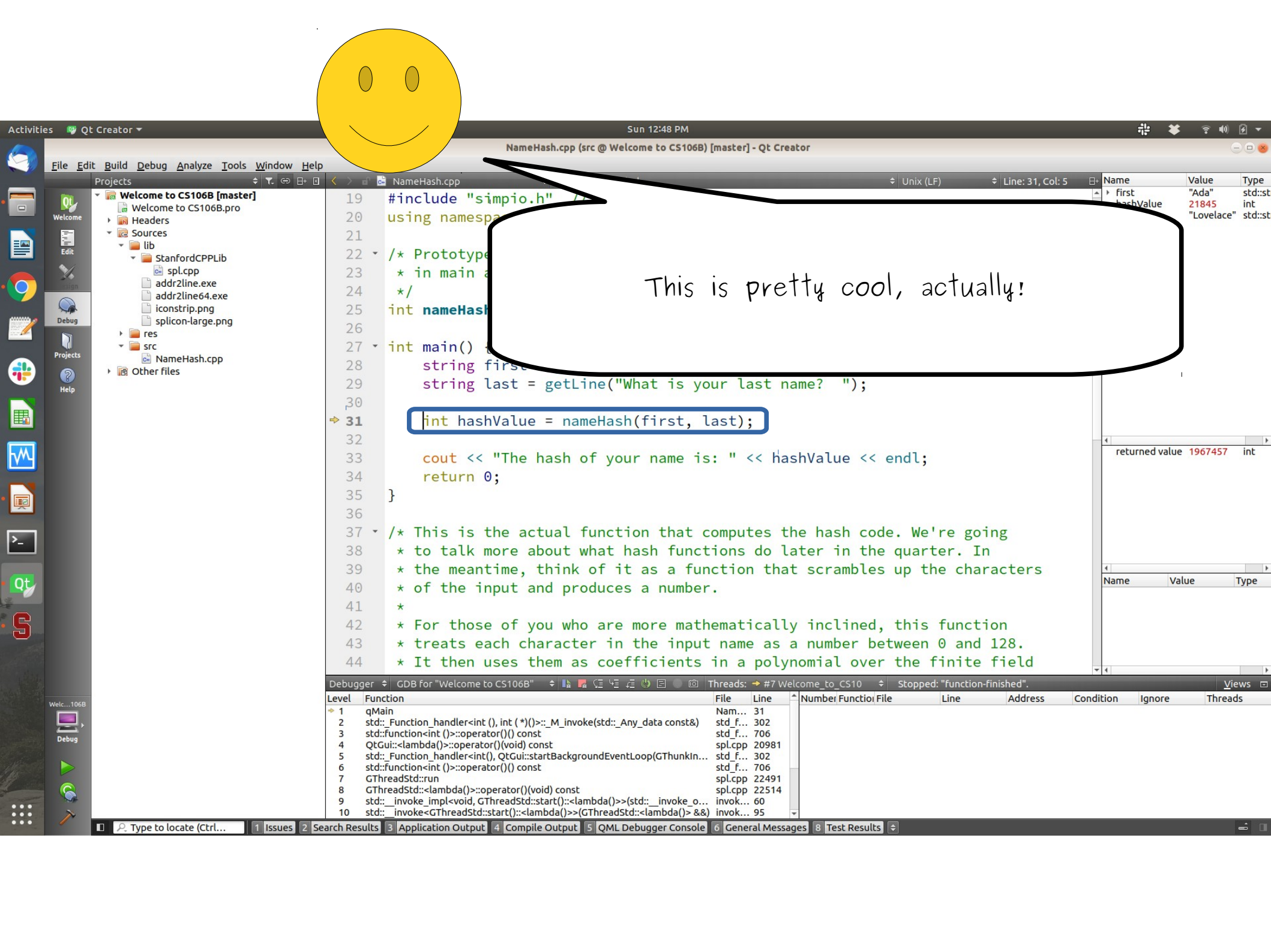

Activities
Qt Creator
Sun 12:48 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 5
#include "simpio.h" // for getLine
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);
cout << "The hash of your name is: " << hashValue << endl;
return 0;
}
/* This is the actual function that computes the hash
* to talk
* the
* of th
* For t
* treat
* It th
Name Value Type
first "Ada" std::st
hashValue 21845 int
last "Lovelace" std::st
1967457 int
Name Value Type
Let's take a minute to get our bearings.
Where exactly are we?
Debugger
GDB for "Welc
Views
Level Function
1 qMain
2 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&)
3 std::function<int ()>::operator()() const
4 QtGui::<lambda()>::operator()(void) const
5 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn...
6 std::function<int ()>::operator()() const
7 GThreadStd::run
8 GThreadStd::<lambda()>::operator()(void) const
9 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o...
10 std::__invoke<GThreadStd::start()::<lambda()>>(GThreadStd::<lambda()> &&)
std_f... 302
std_f... 706
spl.cpp 20981
std_f... 302
std_f... 706
spl.cpp 22491
spl.cpp 22514
invok... 60
invok... 95
Condition Ignore Threads
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

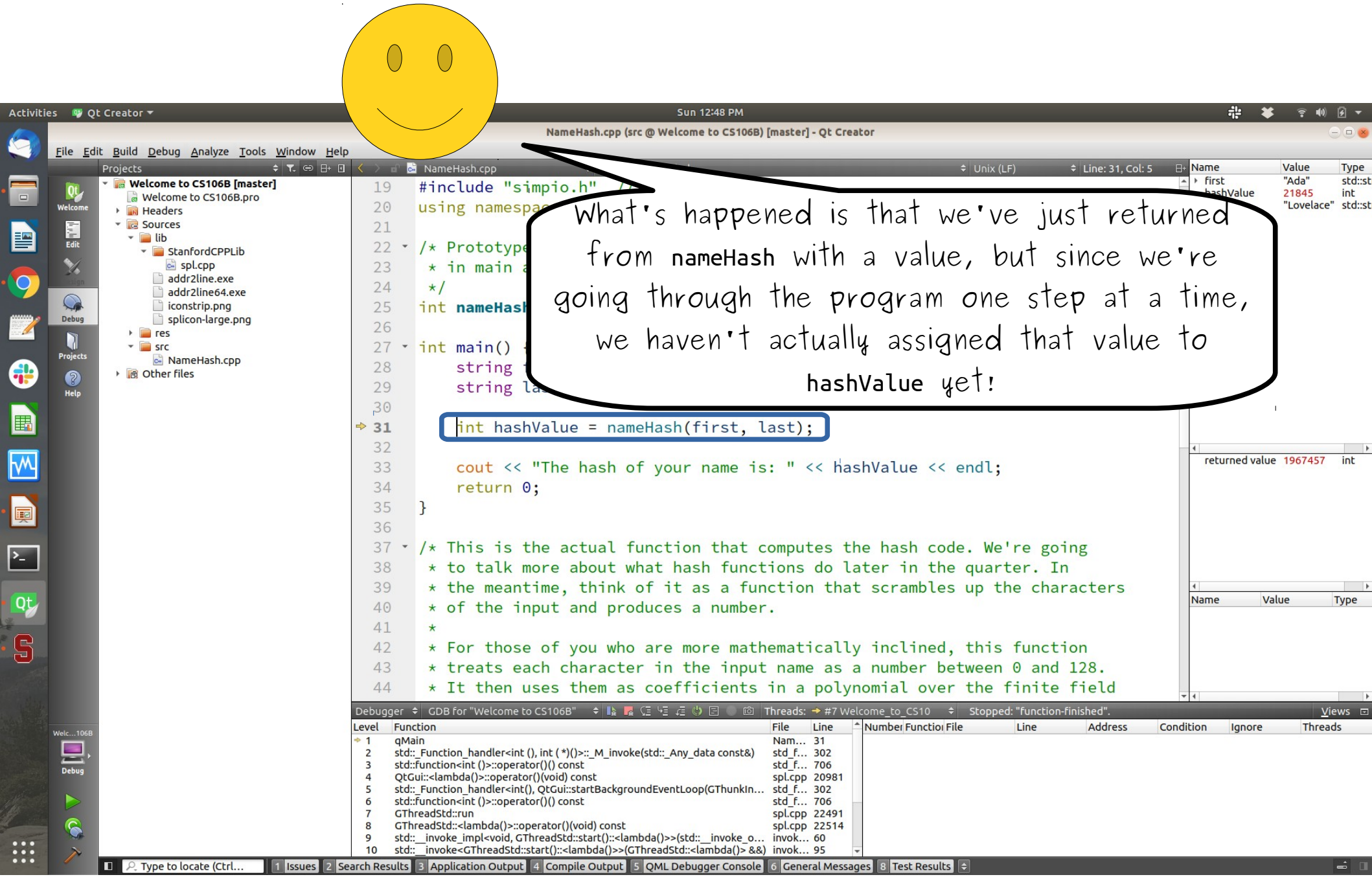

Activities
Qt Creator
Sun 12:48 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 5
#include "simpio.h" // for getLine
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);
cout << "The hash of your name is: " << hashValue << endl;
return 0;
}
/* This is the actual function that computes the hash
* to talk
* the
* of th
*
* For t
* treat
* It th
Name Value Type
first "Ada" std::st
hashValue 21845 int
last "Lovelace" std::st
1967457 int
Name Value Type
Well, the yellow arrow indicates that we're back in main again. Cool!
Debugger
GDB for "Wel
Views
Level Function
1 qMain
2 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&)
3 std::function<int ()>::operator()() const std_f... 706
4 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
5 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn... std_f... 302
6 std::function<int ()>::operator()() const std_f... 706
7 GThreadStd::run spl.cpp 22491
8 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
9 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o... invok... 60
10 std::__invoke<GThreadStd::start()::<lambda()>>(GThreadStd::<lambda()> &&) invok... 95
Condition Ignore Threads
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results


Activities
Qt Creator
Sun 12:48 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
NameHash.cpp
<Select Symbol>
Unix (LF)
Line: 31, Col: 5
#include "simpio.h" // for getLine
using namespace std;
/* Prototype for the nameHash function. This lets us use the function
* in main and then define it later in the program.
*/
Please note: This information in the return value panel may not appear on all systems. If you don't see this value, then just move on to the next step of the tutorial.
string last = getLine("What is your last name? ");
hashValue = nameHash(first, last);
<< "The hash of your name is: " << hashValue << endl;
rn 0;
/* This is the
that computes the hash code. We're going
* to talk
* the
* of th
*
* For t
* treat
* It th
Name Value Type
first "Ada" std::st
hashValue 21845 int
last "Lovelace" std::st
returned value 1967457
We can see that the nameHash function returned 1967457. Thanks, debugger!
Debugger
GDB for "Wel
Views
Level Function
1 qMain
2 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&)
3 std::function<int ()>::operator()() const
4 QtGui::<lambda()>::operator()(void) const
5 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn...
6 std::function<int ()>::operator()() const
7 GThreadStd::run
8 GThreadStd::<lambda()>::operator()(void) const
9 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o...
10 std::__invoke<GThreadStd::start()::<lambda()>>(GThreadStd::<lambda()> &&)
Ignore Threads
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results


Activities
Qt Creator
Sun 12:48 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
<Select Symbol>
Unix (LF)
Line: 31, Col: 5
Name
Value
Type
hashValue
21845
But if you look up over here in the values window, you can see that hashValue has some really weird-looking number stored in it. (You'll almost certainly see something different on your system.)
"Really weird-looking number" is a relative term here, as the number will vary from computer to computer and may be as innocent looking as "0" or "1" or as wild as "2349980890".
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped: "function-finished".
Views
Level
Function
File
Line
Address
Condition
Ignore
Threads
Type to locate (Ctrl...
Issues
Search Results
Application Output
Compile Output
QML Debugger Console
General Messages
Test Results


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


This is pretty cool, actually!


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!

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

Activities Qt Creator Sun 12:48 PM

NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- Welcome to CS106B [master]
  - Welcome to CS106B.pro
  - Headers
  - Sources
    - lib
      - StanfordCPPLib
        - spl.cpp
      - addr2line.exe
      - addr2line64.exe
      - iconstrip.png
      - splicon-large.png
    - res
    - src
      - NameHash.cpp
  - Other files

NameHash.cpp — Unix (LF) — Line: 31, Col: 5

```
19  #include "simpio.h"
20  using namesp
21
22  /* Prototype
23   * in main a
24   */
25  int nameHash
26
27  int main() {
28      string first = getLine("What is your first name? ");
29      string last = getLine("What is your last name?  ");
30
31      int hashValue = nameHash(first, last);
32
33      cout << "The hash of your name is: " << hashValue << endl;
34      return 0;
35  }
36
37  /* This is the actual function that computes the hash code. We're going
38   * to talk more about what hash functions do later in the quarter. In
39   * the meantime, think of it as a function that scrambles up the characters
40   * of the input and produces a number.
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 128.
44   * It then uses them as coefficients in a polynomial over the finite field
```

Name	Value	Type
first	"Ada"	std::st
hashValue	21845	int
	"Lovelace"	std::st

returned value 1967457 int

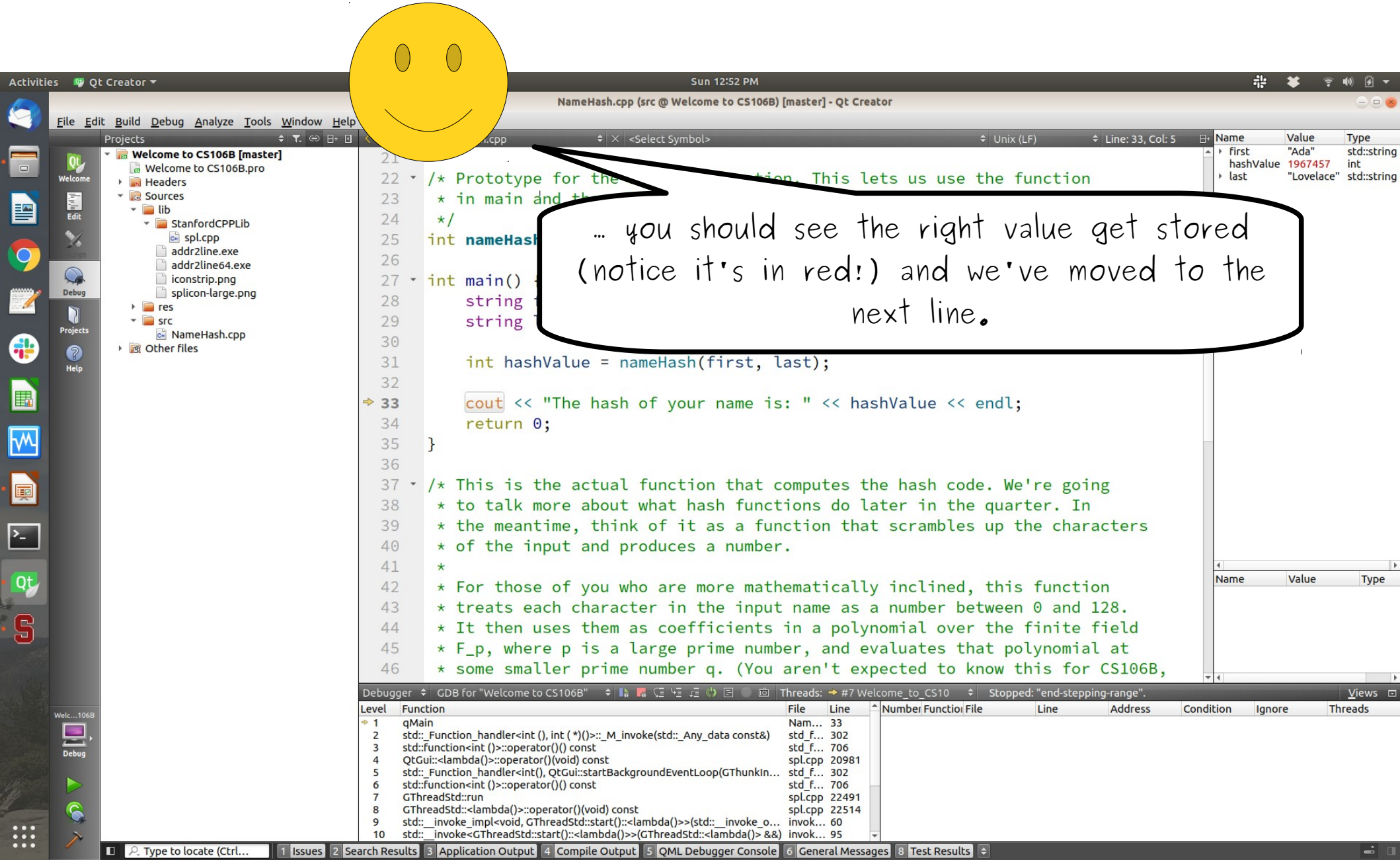
Name	Value	Type

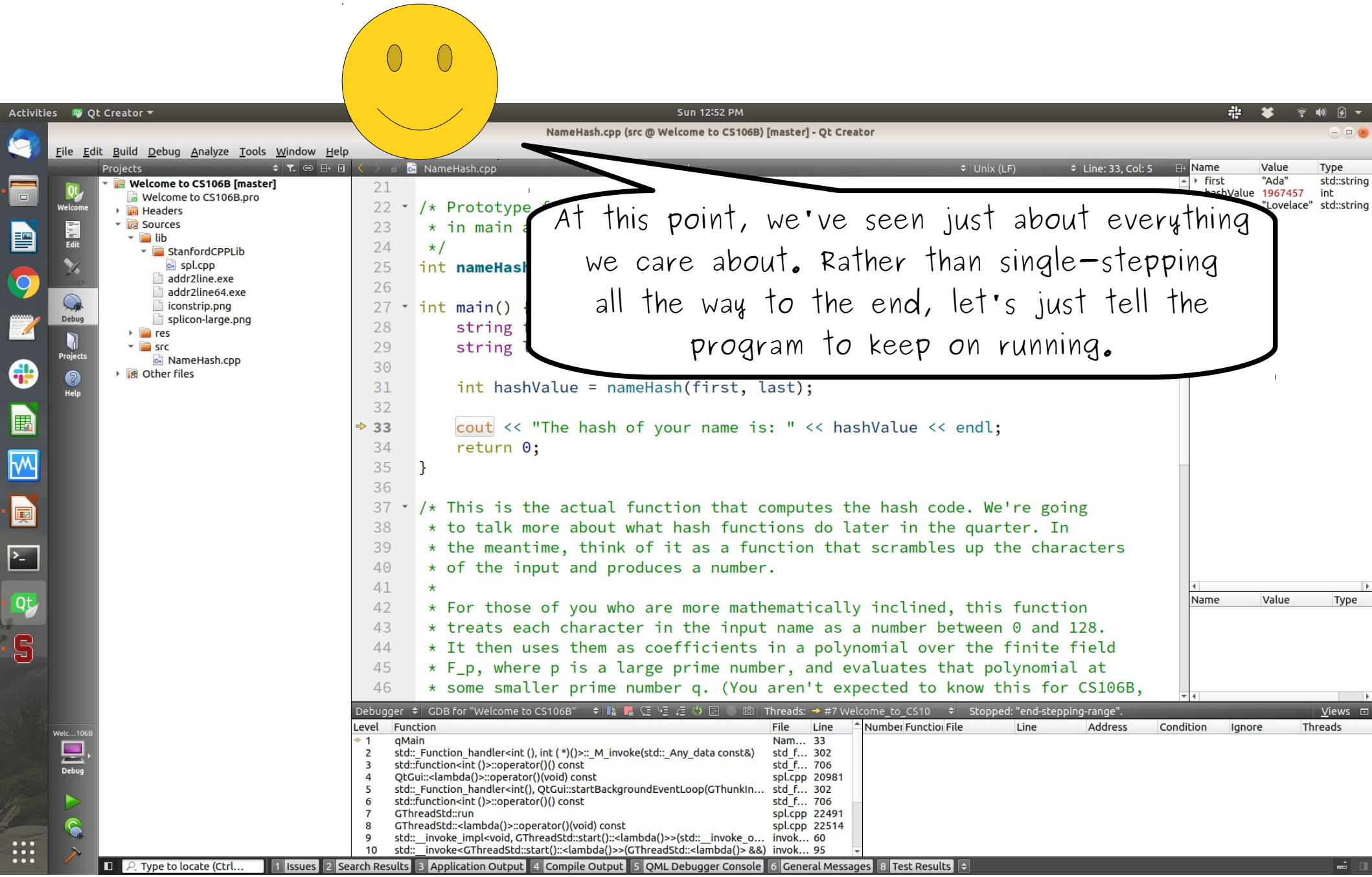
Debugger — GDB for "Welcome to CS106B" — Threads: #7 Welcome_to_CS10 — Stopped: "function-finished". — Views

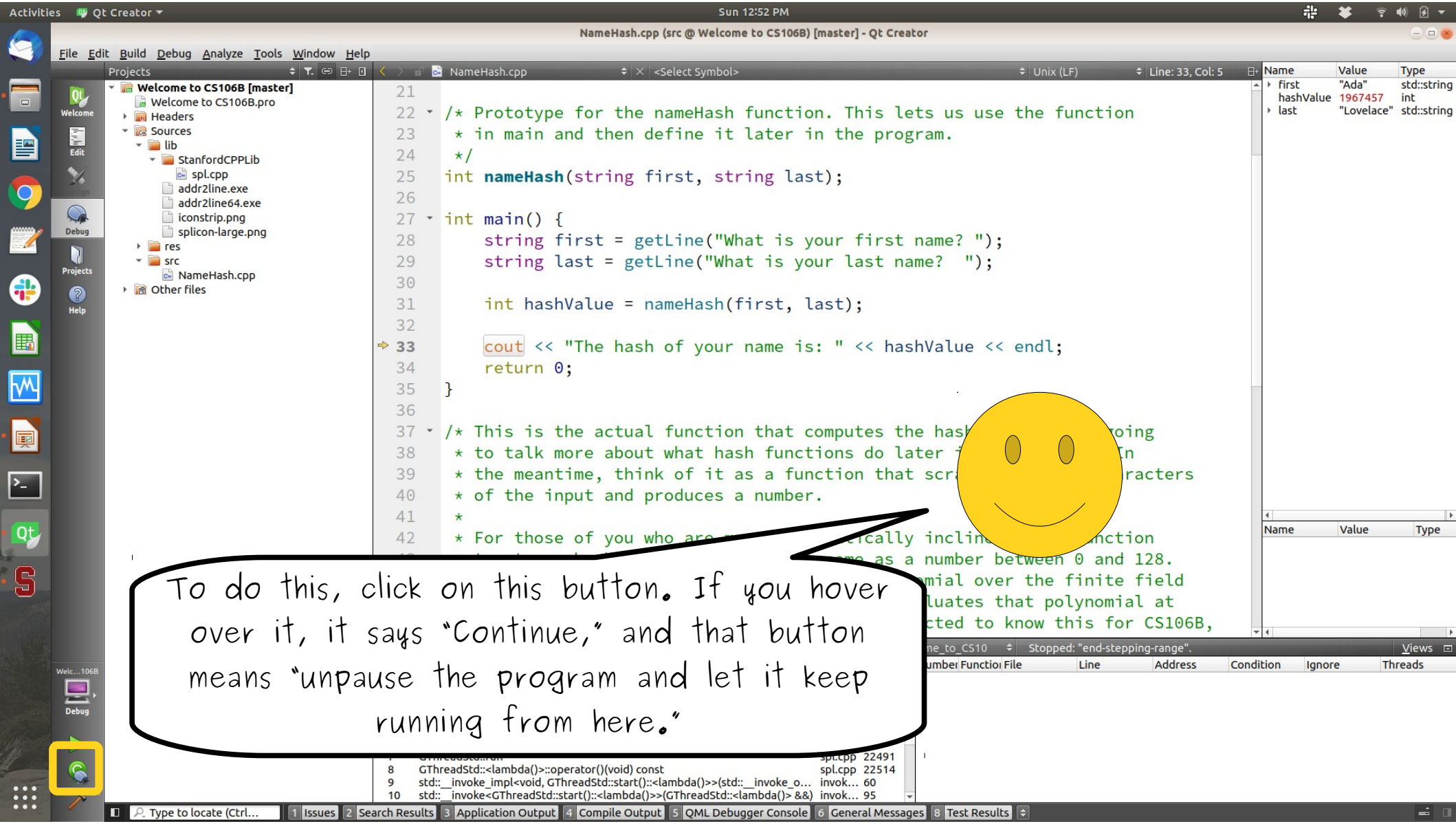
Level	Function	File	Line
1	qMain	Nam…	31
2	std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&)	std_f…	302
3	std::function<int ()>::operator()() const	std_f…	706
4	QtGui::<lambda()>::operator()(void) const	spl.cpp	20981
5	std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn…	std_f…	302
6	std::function<int ()>::operator()() const	std_f…	706
7	GThreadStd::run	spl.cpp	22491
8	GThreadStd::<lambda()>::operator()(void) const	spl.cpp	22514
9	std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o…	invok…	60
10	std::__invoke<GThreadStd::start()::<lambda()>>(GThreadStd::<lambda()> &&)	invok…	95

Number	Functio	File	Line	Address	Condition	Ignore	Threads

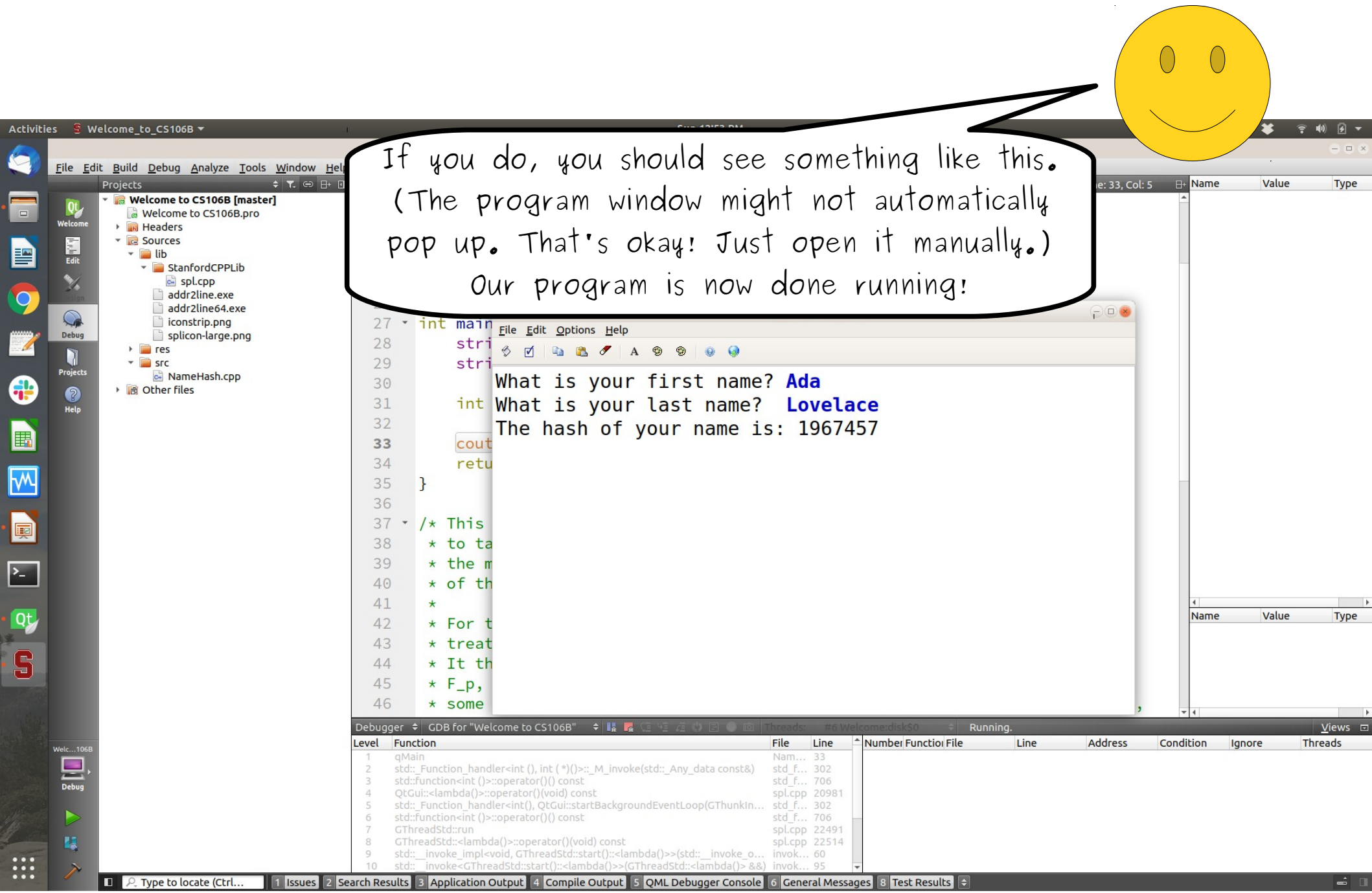
Type to locate (Ctrl… 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results


Activities
Qt Creator
Sun 12:52 PM
NameHash.cpp (src @ Welcome to CS106B) [master] - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome to CS106B [master]
Welcome to CS106B.pro
Headers
Sources
lib
StanfordCPPLib
spl.cpp
addr2line.exe
addr2line64.exe
iconstrip.png
splicon-large.png
res
src
NameHash.cpp
Other files
<Select Symbol>
Unix (LF)
Line: 33, Col: 5
/* Prototype for the
tion. This lets us use the function
* in main and th
*/
int nameHash
int main()
string
string
int hashValue = nameHash(first, last);
cout << "The hash of your name is: " << hashValue << endl;
return 0;
}
/* This is the actual function that computes the hash code. We're going
* to talk more about what hash functions do later in the quarter. In
* the meantime, think of it as a function that scrambles up the characters
* 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 128.
* It then uses them as coefficients in a polynomial over the finite field
* F_p, where p is a large prime number, and evaluates that polynomial at
* some smaller prime number q. (You aren't expected to know this for CS106B,
Name Value Type
first "Ada" std::string
hashValue 1967457 int
last "Lovelace" std::string
… you should see the right value get stored (notice it's in red!) and we've moved to the next line.
Debugger
GDB for "Welcome to CS106B"
Threads:
#7 Welcome_to_CS10
Stopped: "end-stepping-range".
Views
Level Function File Line
1 qMain Nam… 33
2 std::_Function_handler<int (), int (*)()>::_M_invoke(std::_Any_data const&) std_f… 302
3 std::function<int ()>::operator()() const std_f… 706
4 QtGui::<lambda()>::operator()(void) const spl.cpp 20981
5 std::_Function_handler<int(), QtGui::startBackgroundEventLoop(GThunkIn… std_f… 302
6 std::function<int ()>::operator()() const std_f… 706
7 GThreadStd::run spl.cpp 22491
8 GThreadStd::<lambda()>::operator()(void) const spl.cpp 22514
9 std::__invoke_impl<void, GThreadStd::start()::<lambda()>>(std::__invoke_o… invok… 60
10 std::__invoke<GThreadStd::start()::<lambda()>>(GThreadStd::<lambda()> &&) invok… 95
Number Functio File Line Address Condition Ignore Threads
Type to locate (Ctrl…
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results

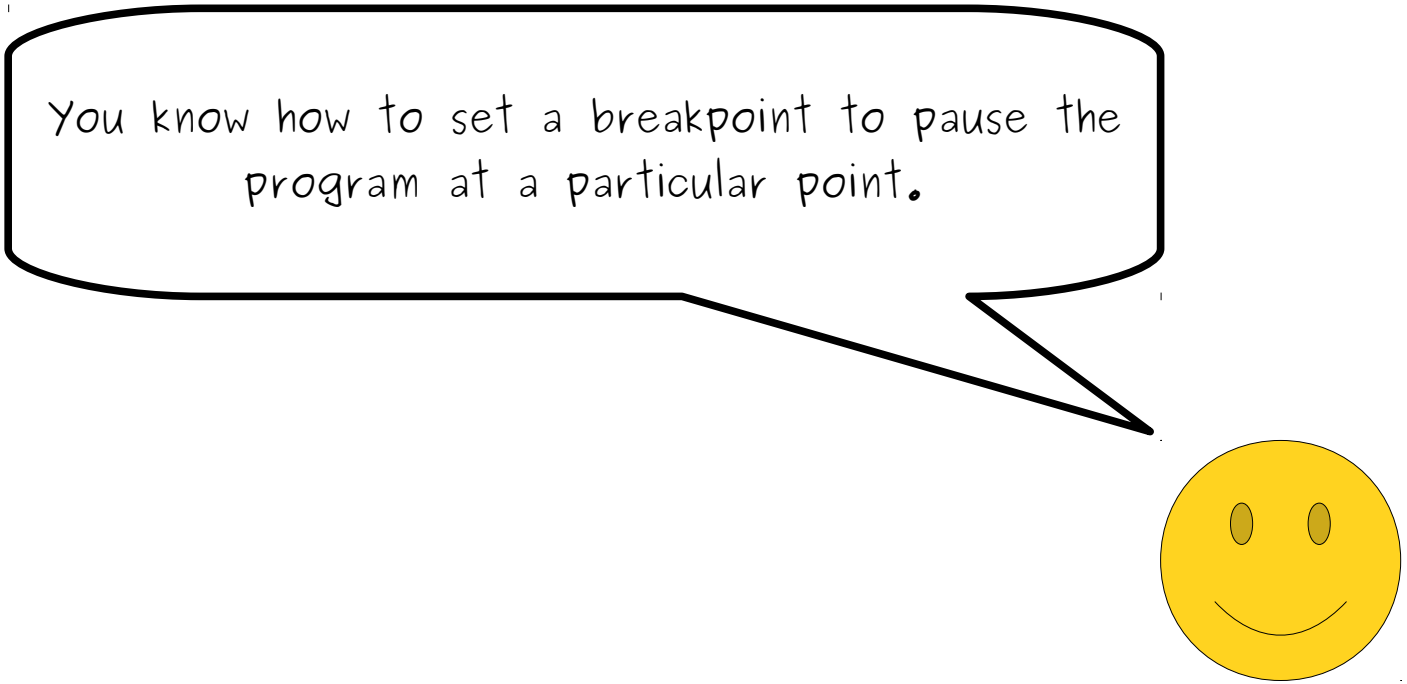

At this point, we've seen just about everything we care about. Rather than single-stepping all the way to the end, let's just tell the program to keep on running.

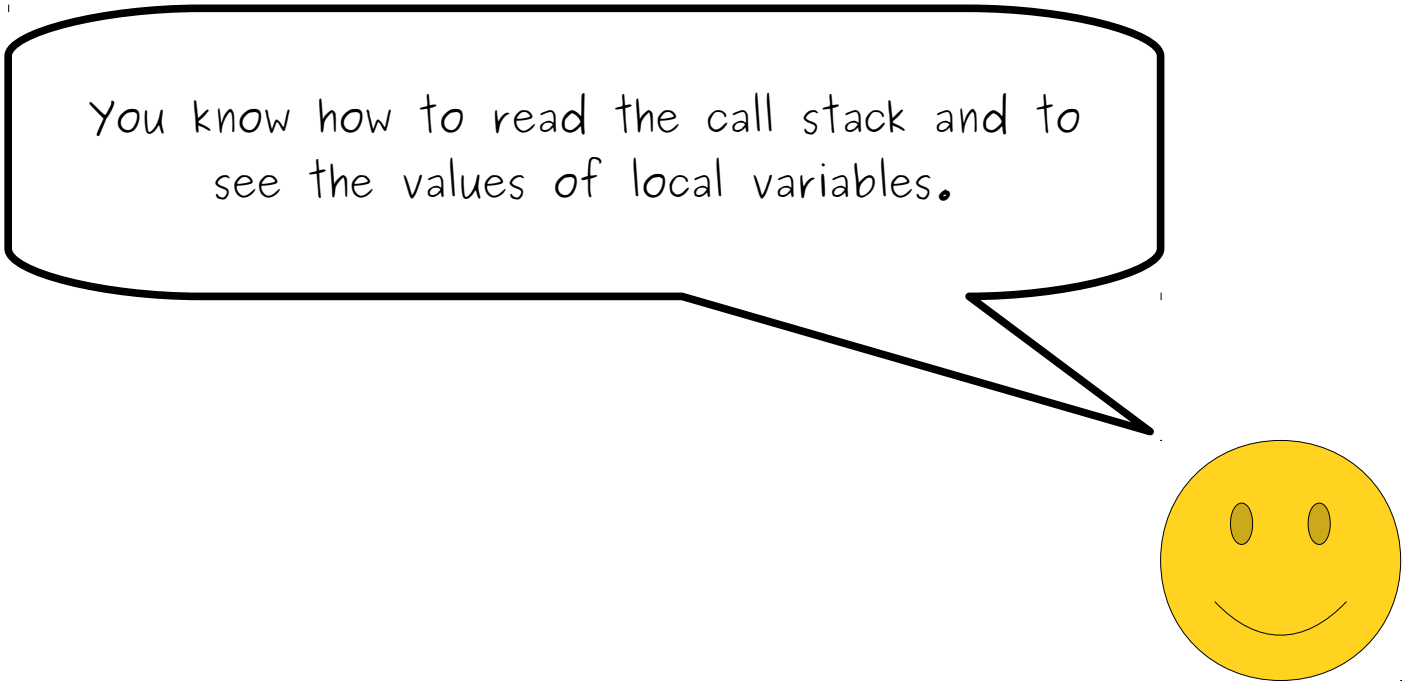


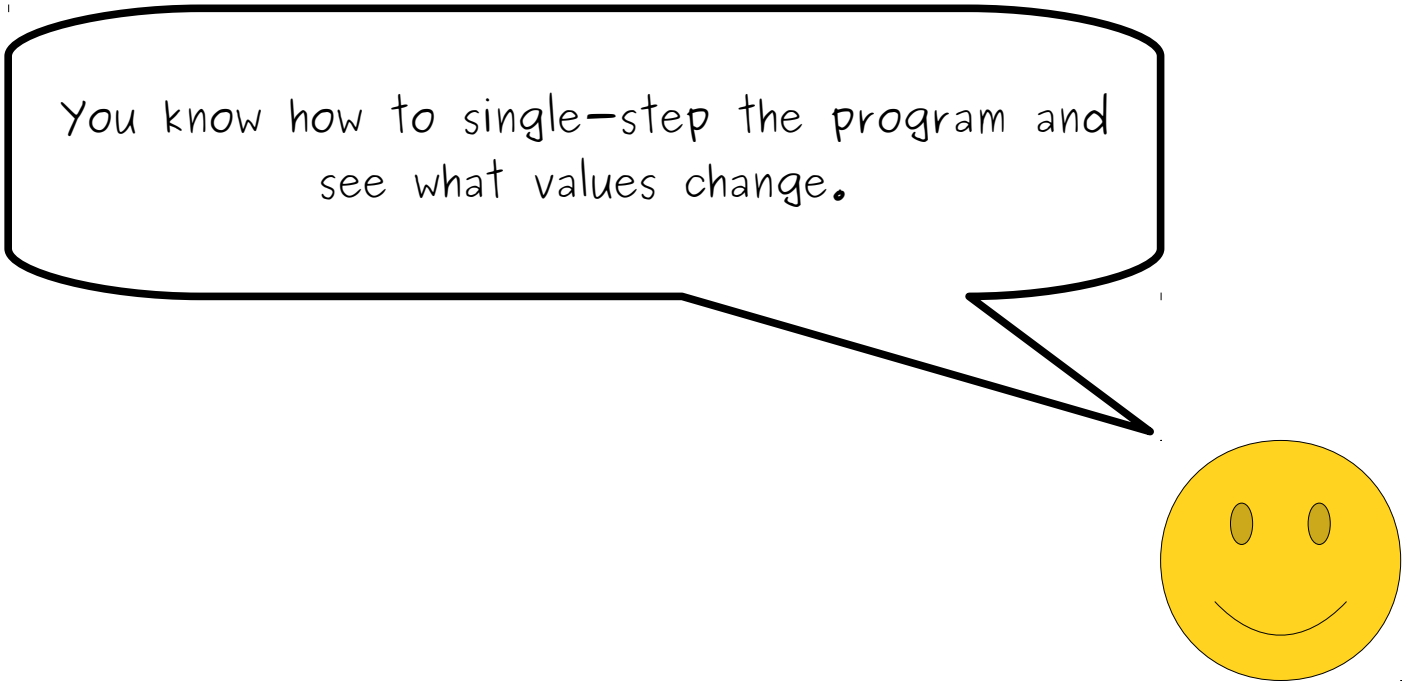
To do this, click on this button. If you hover over it, it says "Continue," and that button means "unpause the program and let it keep running from here."

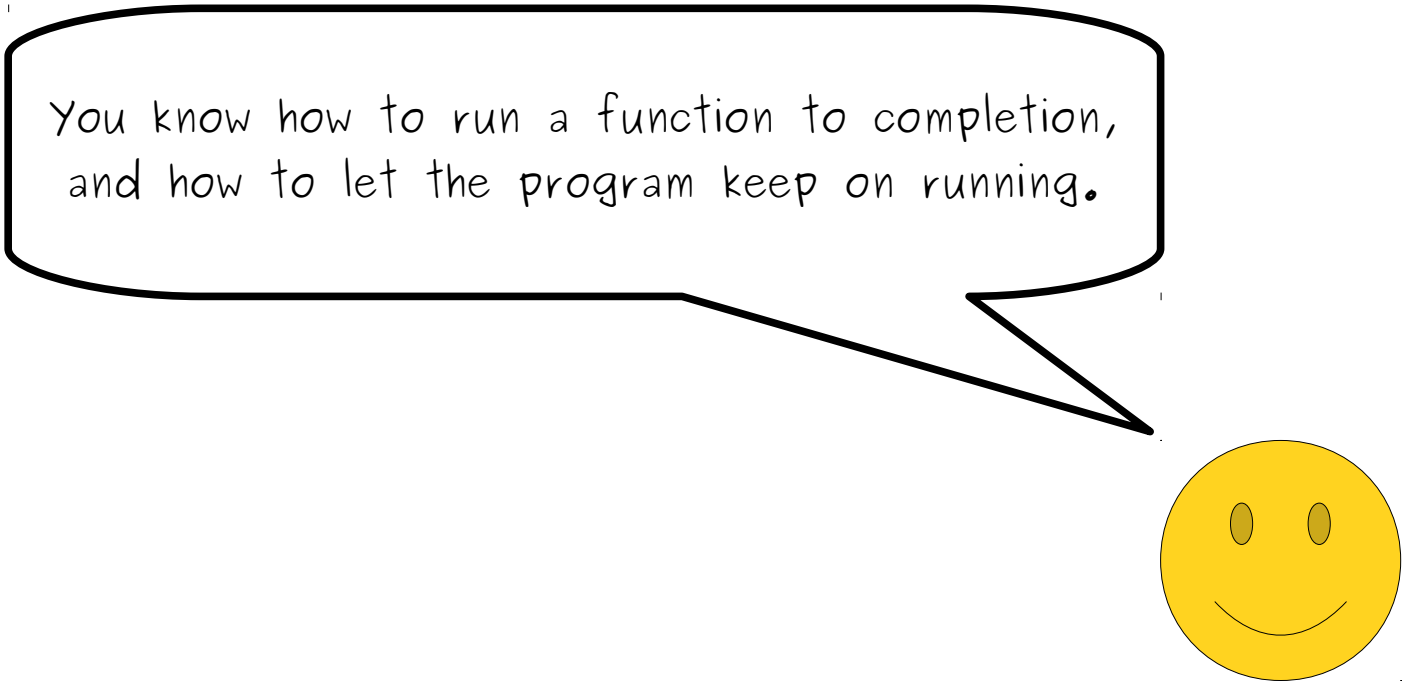

If you do, you should see something like this.
(The program window might not automatically pop up. That's okay! Just open it manually.)
Our program is now done running!
What is your first name? Ada
What is your last name? Lovelace
The hash of your name is: 1967457

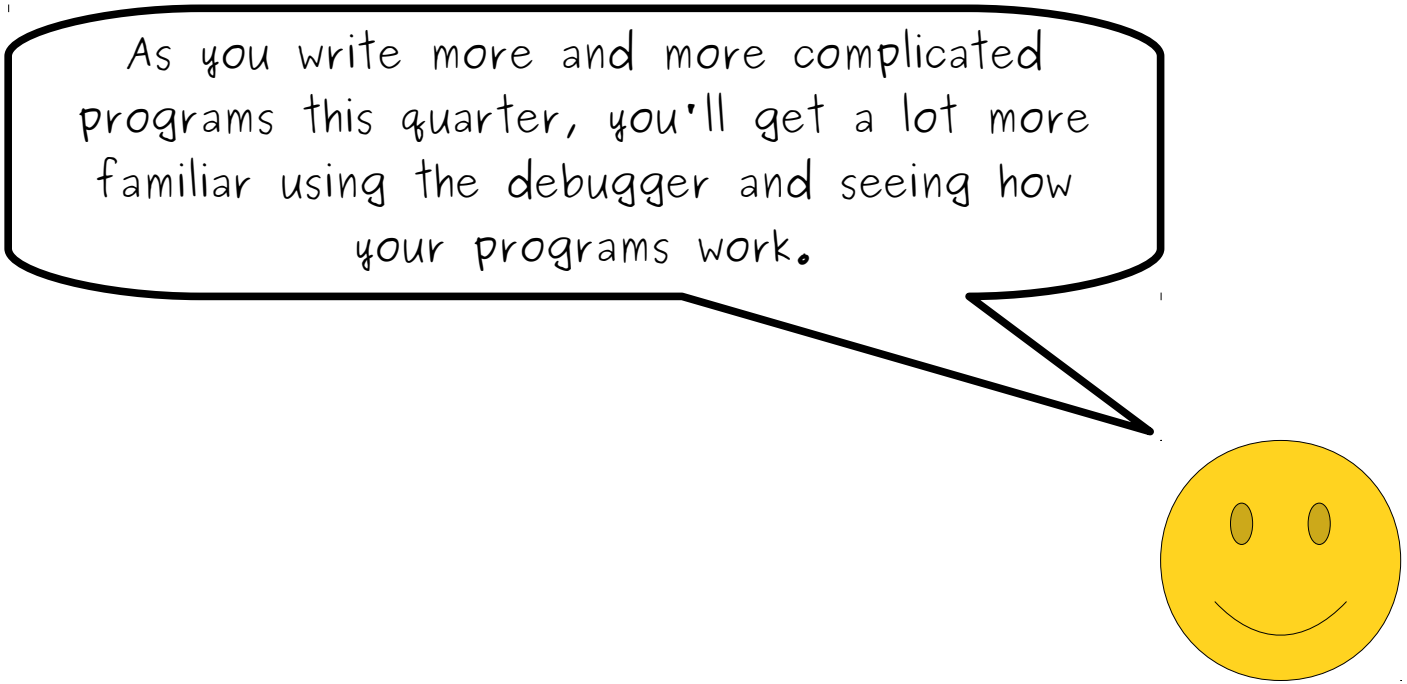

So there you have it! You've now gotten more familiar with the debugger!

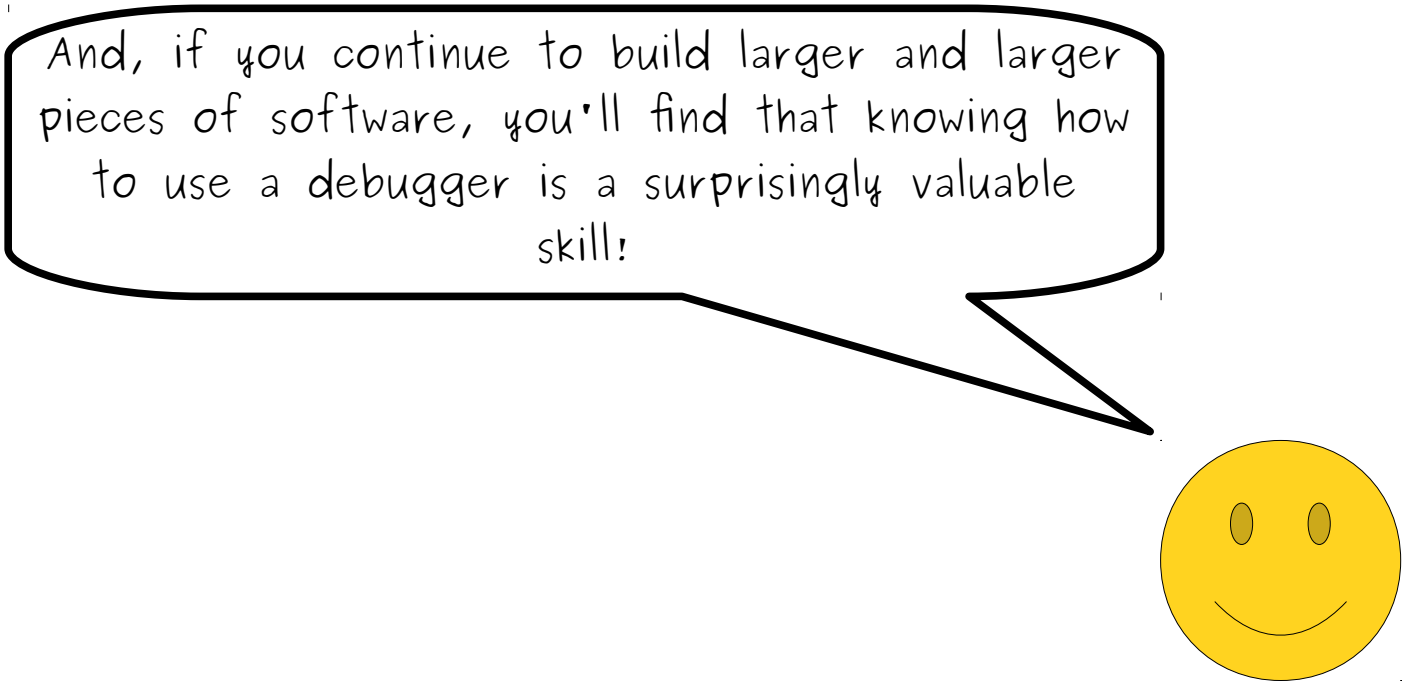

You know how to set a breakpoint to pause the program at a particular point.


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


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


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


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


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


Hope this helps, and welcome to CS106B!