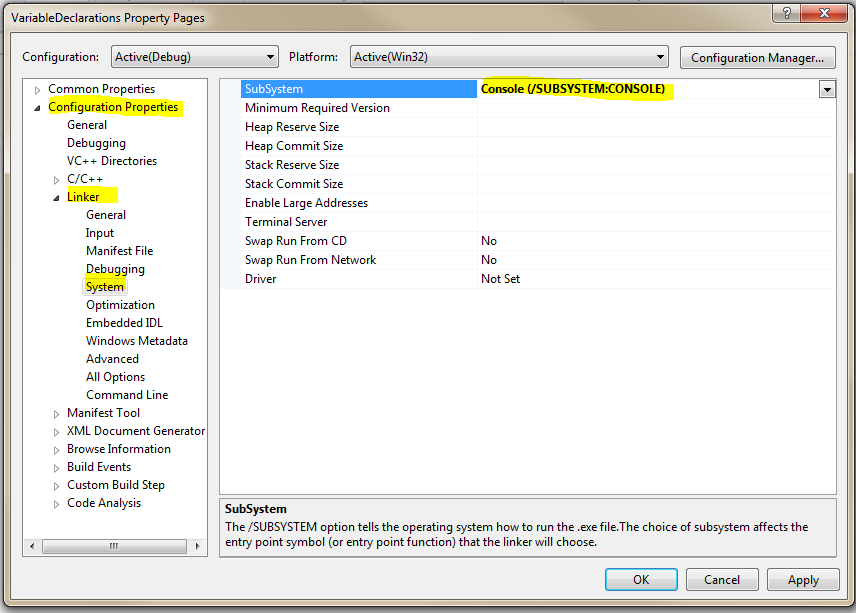
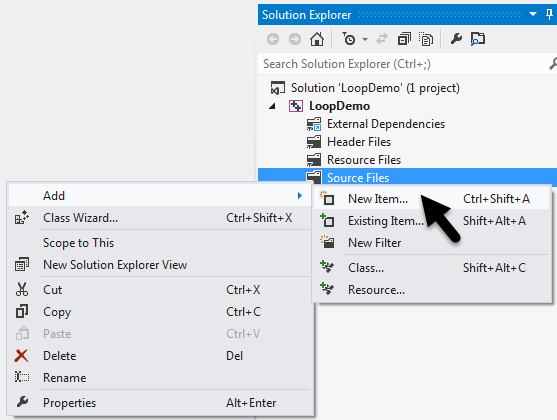
Demo Passing arrays

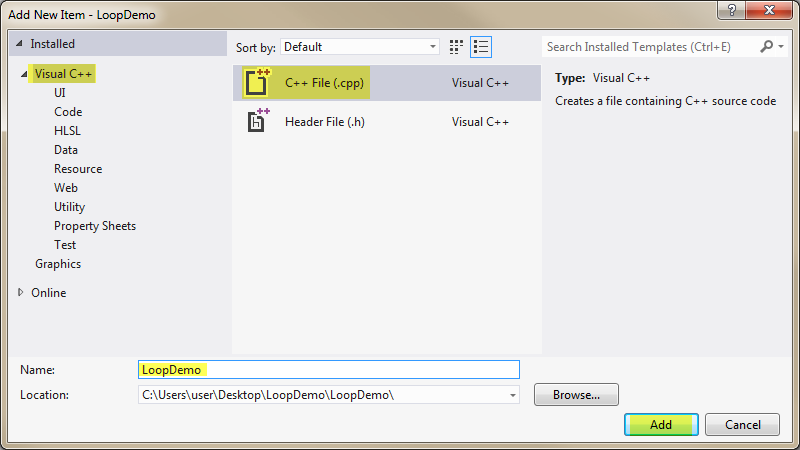
1. Start our program
   1. Start Visual Studio
   2. Create a new empty C++ project:
   3. Call it “PassingArraysDemo”, pick the desktop as the location and click OK.
   4. Once your solution is created make sure your project is set up to run without closing the console window. Right click on the project (Called “PassingArraysDemo”) and select “properties”.
   5. In the dialog window select Configuration Properties|Linker|System and set SubSystem to “Console(/SUBSYSTEM:CONSOLE)”:



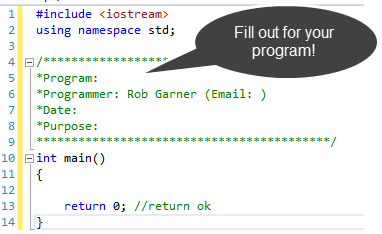
* 1. Click on Source Files and Add|New Item:



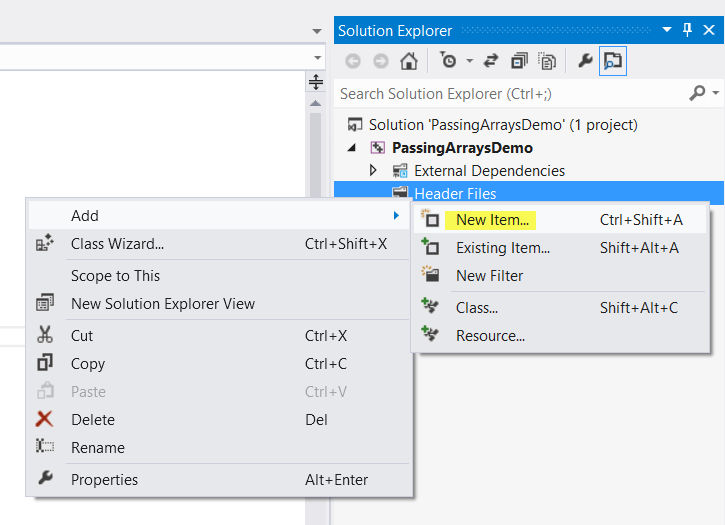
* 1. Select Visual C++, C++ File (.cpp), call it “PassingArraysDriver, then click “Add”:



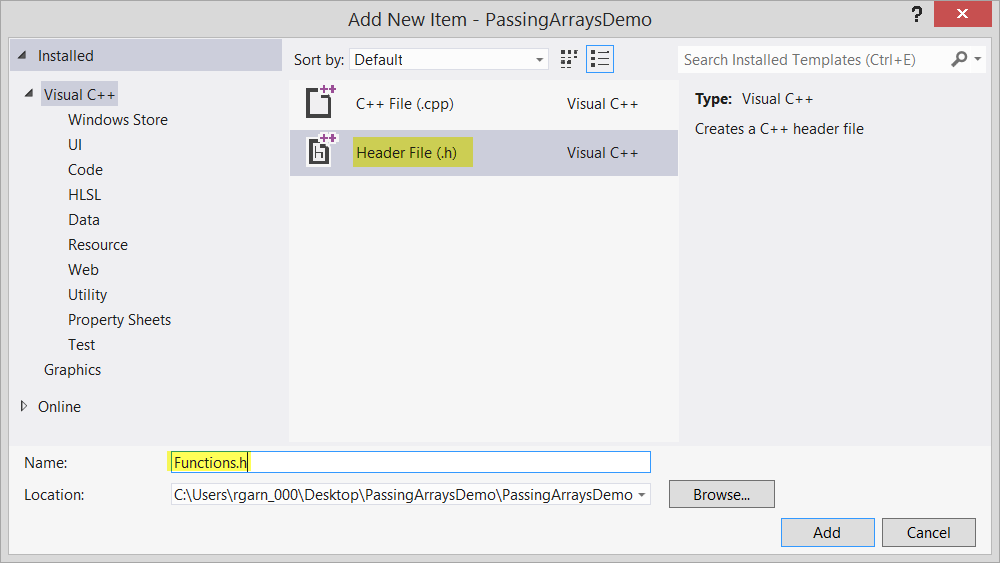
* 1. Enter the following code to start our program:



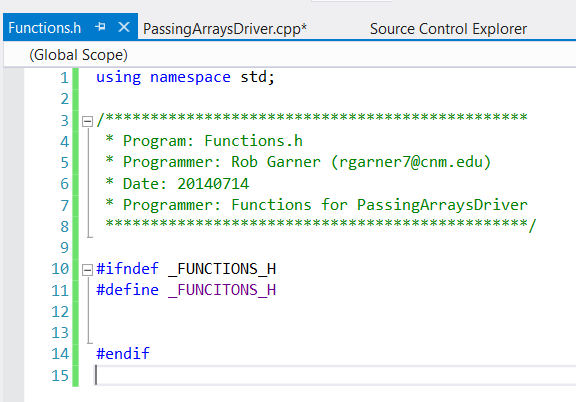
1. Create a function that fills an array with random numbers
   1. Right click on your project’s Header Files directory and add a new item:



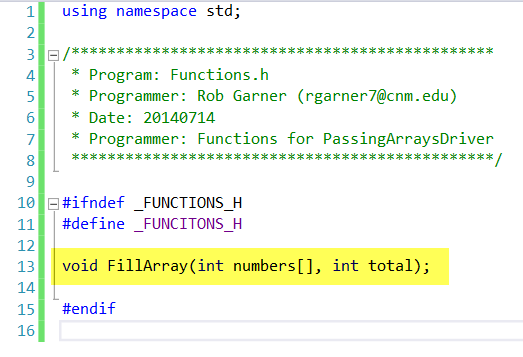
* 1. Add a Functions.h file:



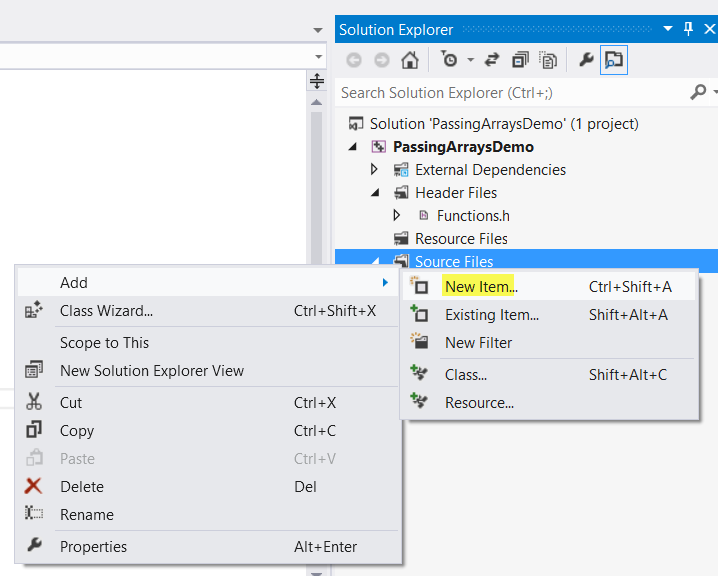
* 1. Enter the usual items we need for a header file:



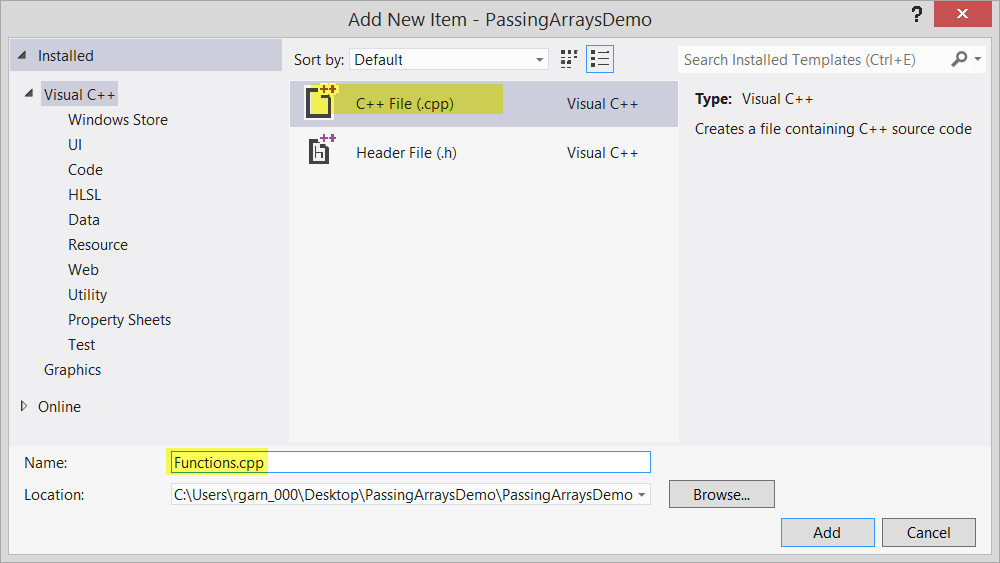
* 1. Add the following function prototype:



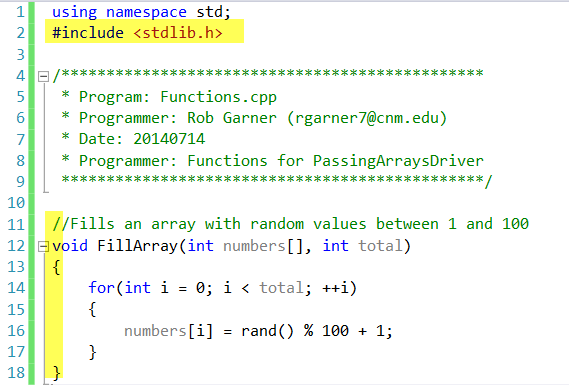
* 1. Add a new item to Source Files:



* 1. Add a Functions.cpp file:

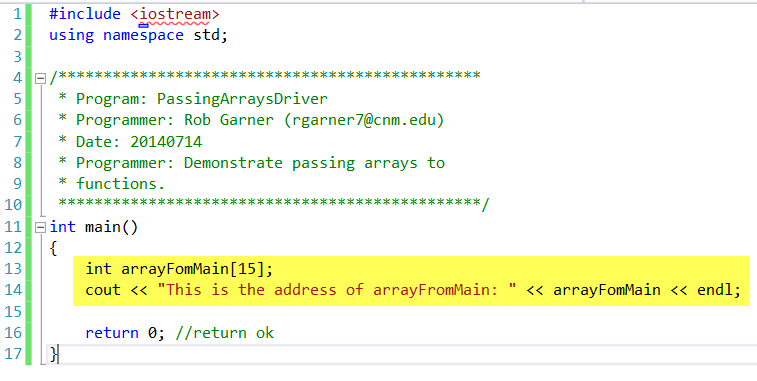


* 1. Add a fill array function

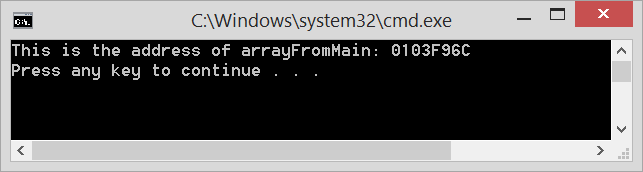


* 1. Notice that we have to include stdlib.
  2. Notice that we don’t need to use the \* operator for the array in the parameter list or when setting values into the array!

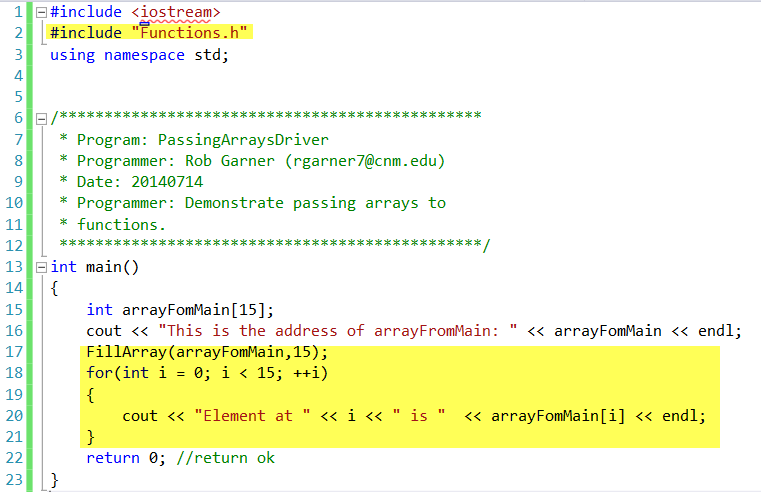
1. Display all the numbers
   1. In PassingArraysDriver.cpp add the following code:



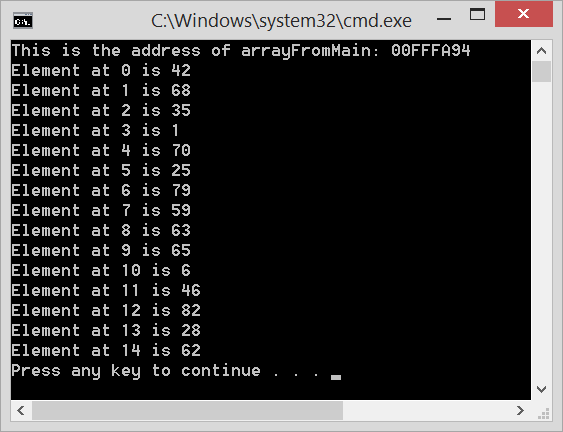
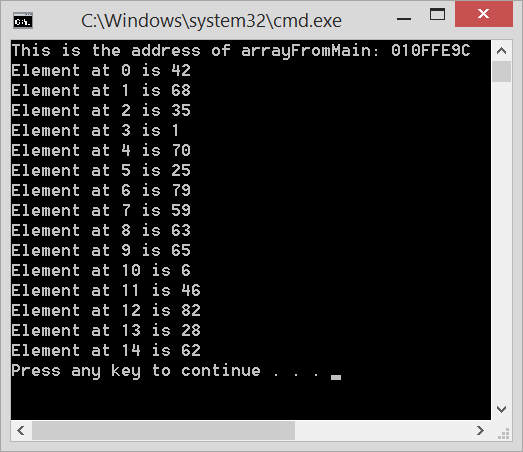
* 1. Try it with Ctrl-F5:



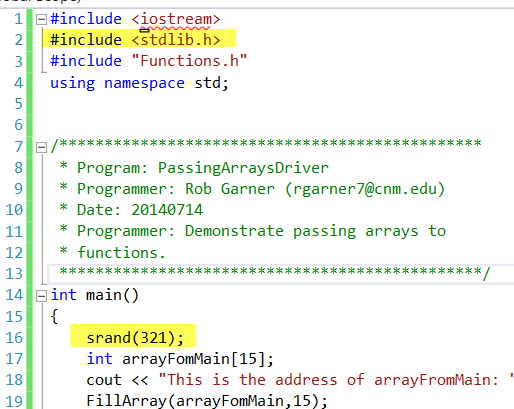
* 1. Notice the name of the array is really just an address! In other words it’s a pointer.
  2. Change the code to:



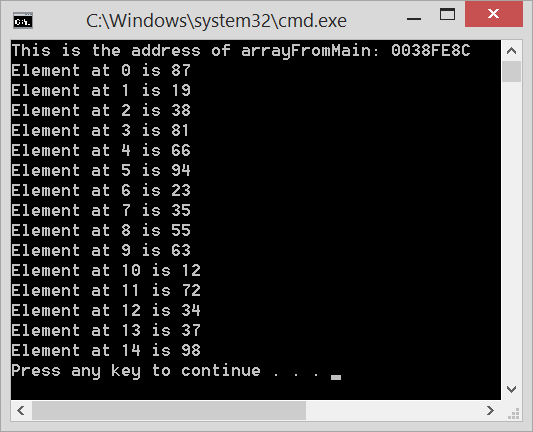
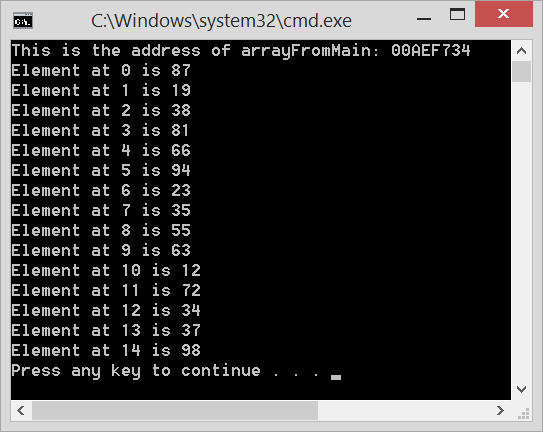
* 1. Notice that we now include our Funcitons.h file and also display each element.
  2. Notice that we don’t need to use the & operator when passing the array!
  3. Try it two times with Ctrl-F5

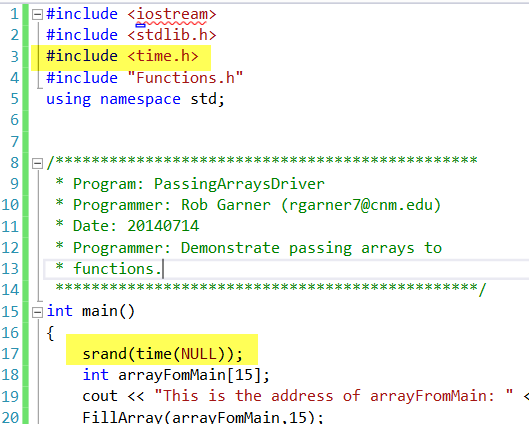
* 1. Notice that the location of the array changes each time but that the sequence of numbers is the same. We need to seed the random number generator!
  2. Change PassingArraysDriver to:



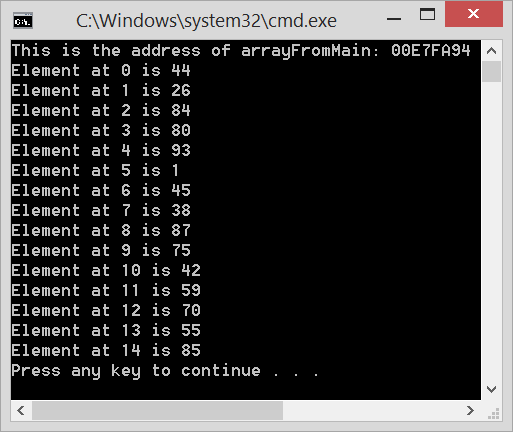
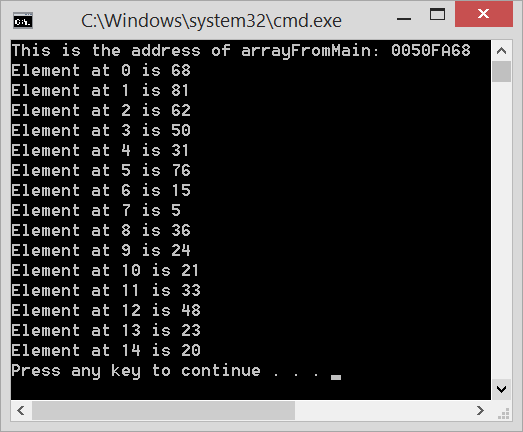
* 1. Try it two times with Ctrl-F5

* 1. Again notice we get the same sequence in both runs but this time it is a different sequence.
  2. Change the code again:

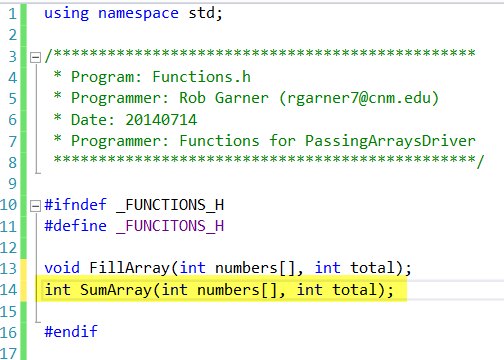


* 1. Notice we now add the time library and seed using the current time by calling the time(NULL) function. NULL is an empty value that causes us to just get the current system time.
  2. Try it two times with Ctrl-F5

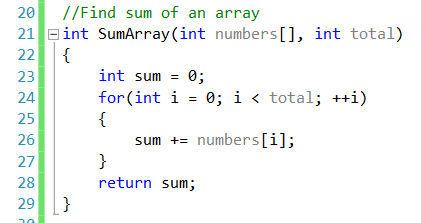
 

* 1. Notice this time that the results change with each run.

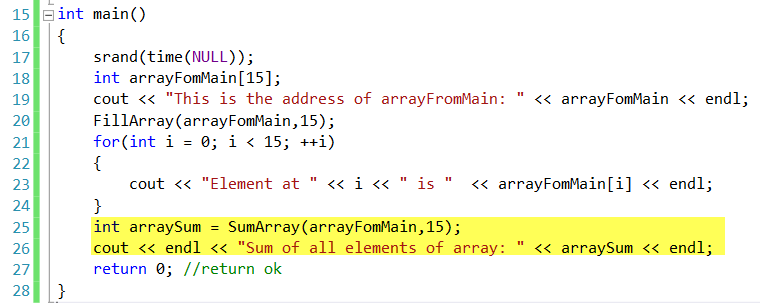
1. Create another function that finds the sum of all items in the array
   1. Add another function prototype to Functions.h:



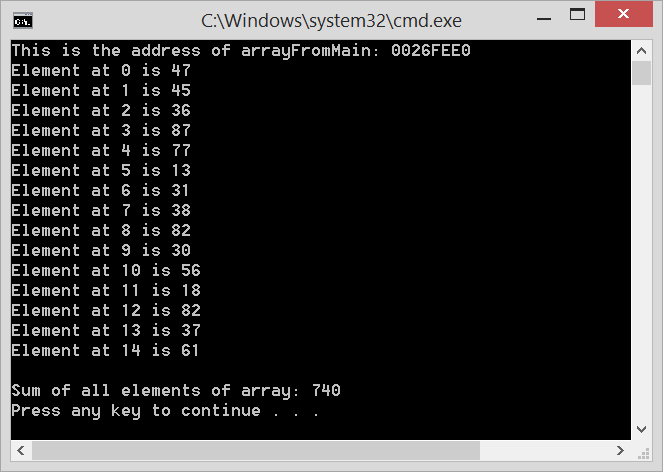
* 1. Again notice that we don’t need to use the & operator when passing the array!
  2. Add the following function to Functions.cpp:



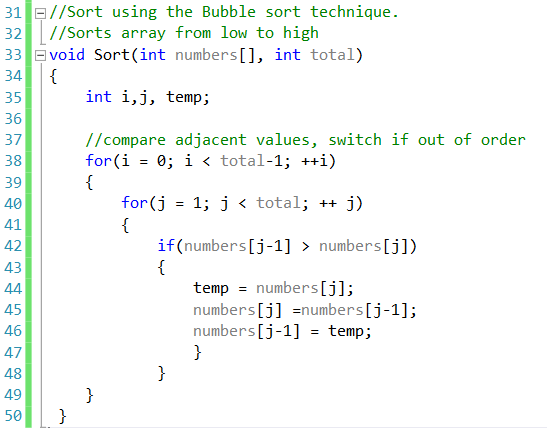
1. Displays the sum
   1. Add the following code to PassingArraysDriver.cpp:



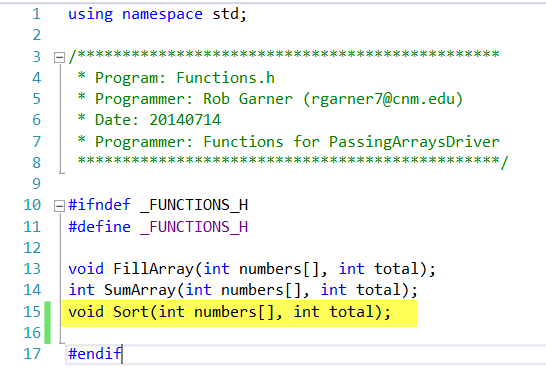
* 1. Notice that we don’t need to use the & operator when passing the array!
  2. Try it with Ctrl-F5:



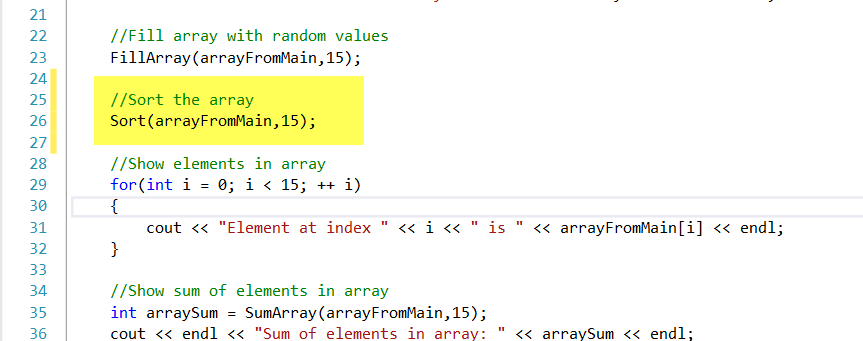
1. Sort the elements in the array
   1. Add the following code to Functions.cpp:



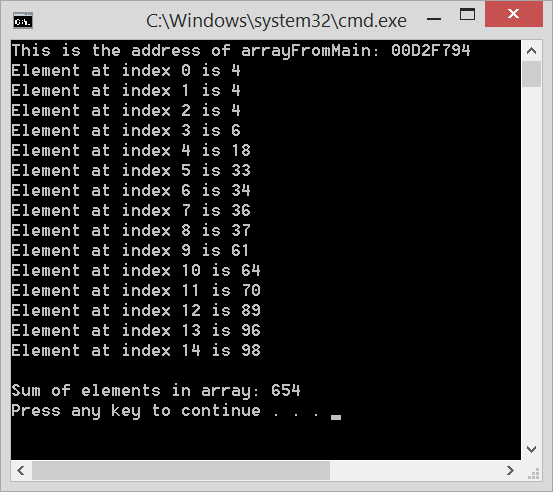
* 1. Add the function prototype to Functions.h:



* 1. Add the following code to main:



* 1. Try it with Ctrl-F5



* 1. Notice all the elements of the array are now sorted!