Computer Programming Lab 11

T.A : Yoo Yeonil, Jeong Wonil

1. Announcements

1. Due date of Assignment 4 is delayed to Dec 3rd.

2. Description of Problem 3 in assign4 is changed.

3. Assignment 5 will be out soon.

1. Announcements

Most of errors from Clion were from Mingw or Cygwin with Cmake or Makefile

2. Alternatives for c++ environment

1. Code::Blocks

    - https://sourceforge.net/projects/codeblocks/files/Binaries/16.01/Windows/codeblocks-16.01mingw-setup.exe/download

2. Command line

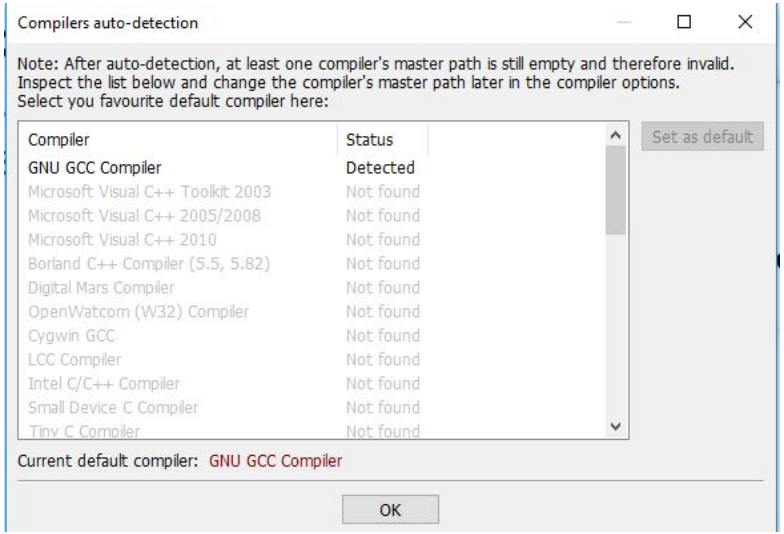
    - This makes writing source code harder, but easier to avoid package problem for submission.

3. Code::Blocks

1. Install code::blocks

2. Run code::blocks and select compiler(GNU gcc will be there)

3. Associate c/c++ type



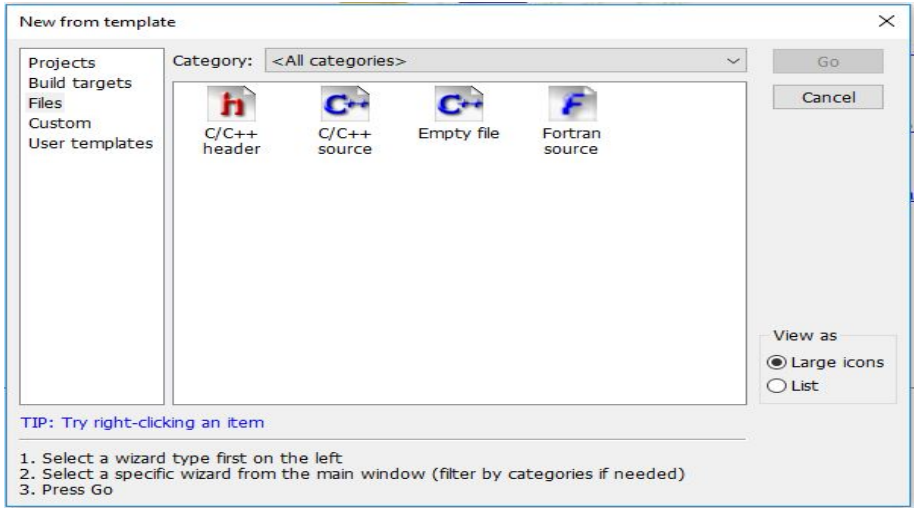
3. Code::Blocks

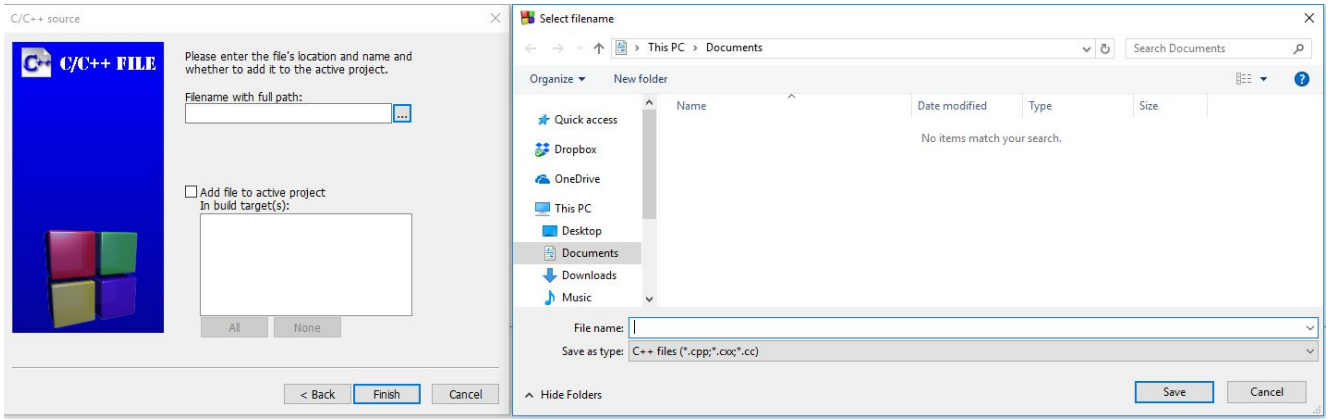
4. New -> file

5. Select C/C++ source file

6. Select C++, and click next.

7. Set the path of your cpp file and its name.





4. Practice

Let’s say a triangle object has 3 arrays and they share first and end elements.

The size of triangle object depends on input.

e.g.) Triangle 1: [1, 20, 38, 21, 0] [0, 2, 3, 20, 111] [111, 20, 33, 21, 1]

Compare two triangle objects by the following rules. Assume triangles from input have the same size and one triangle has 3 arrays with the same size.

- If one data structure’s contents are the same as those of the other rotated or not, they are the same.

4. Practice

e.g.) Triangle 1: [1, 20, 38, 21, 0] [0, 2, 3, 20, 111] [111, 20, 33, 21, 1]

Triangle 2: [0, 2, 3, 20, 111] [111, 20, 33, 21, 1] [1, 20, 38, 21, 0]

=> Triangle 1 and 2 are the same.

Print out whether the two triangle structures are the same or not by printing “True” or “False”. If the given input contains a triangle whose end elements does not match, print “Format Error”.

4. Practice

e.g.) Triangle 1: [1, 20, 38, 21, 0] [7, 2, 3, 20, 111] [111, 20, 33, 21, 1]

Triangle 2: [7, 2, 3, 20, 111] [111, 20, 33, 21, 1] [1, 20, 38, 21, 0] //left rotate

=> Format Error

e.g.) Triangle 1: [1, 2, 3, 4] [4, 3, 2, 1] [1, 2, 3, 1]

Triangle 2: [1, 2, 3, 1] [1, 2, 3, 4] [4, 3, 2, 1] //Right rotate of Triangle 1

=> True

4. Practice

Input example

3	// array size is 3 , Range: 2 ~ 80
1 2 3 3 3 4 4 3 1	//[1,2,3] [3,3,4] [4,3,1] , element Range: 0 ~ 100000
1 3 1 1 2 4 4 3 1	//[1,3,1] [1,2,4] [4,3,1]

output example

False	//True, False, Format Error

4. Practice

Input example

6 // array size is 3 , Range: 2 ~ 80

1 2 3 4 5 1 1 1 3 3 4 1 1 1 100 99 99 1 //[1,2,3,4,5,1][1,1,3,3,4,1][1,1,100,99,99,1]

1 1 100 99 99 1 1 3 3 4 1 1 2 3 4 5 1 //[1,1,100,99,99,1] [1,1,3,3,4,1][1,2,3,4,5,1]

output example

False //True, False, Format Error