## Lab 3

## Part 1 Tile Match

## Part 2 Code Parse

18th November @9.am

Your name and student number must be at the top of every file.

You cannot use <string>or anything else except char ,ints and an array of chars.

Apart from cout, you **can only use functions that you have written**.

You may ask for some help but the code must be substantially yours.

All code must be consistently indented and variables given appropriate names.

Download lab3.h and lab3.cpp from blackboard .

Create a blank project. Add the cpp file and the h file to the project.

Edit lab3.h to write to code for the functions.

You edit lab3.cpp to perform you own tests ( not googletests)

To Upload:

Code (lab3.h file and lab3.cpp)

Lab Book (pdf or doc)

Link to demo video. ( put in comments beside your name on top of cpp file.)

Use onedrive to store the video.

You must upload the code to blackboard before 18th November @9 a.m even if not all the questions are completed.

Upload code , video link and lab book before the deadline.

**You must upload a video link for the**  the lab and **upload the code** (and **Lab Book doc )**to get a mark.

Lab Book Diary

You must also show your lab book, when demoing the code.

The lab book shows your work as you do it and illustrates your approach to solving the problems below. The lab book and demo are worth 50 %.

You must create a  **5 minute video** explaining the code and how you came about the solution or how you tried to come up with a solution.

You can use ObsStudio (https://obsproject.com/ ) to screen record. You should use your rough work and code to explain how you came about your solution.

Contents of Video

Running the code from visual Studio.

Showing the tests you used to check your code.

Explain how you came about your code solution, supported with your code and your lab book. It may include approaches you tried but didn’t work.

**Part 1**

int ReplaceWithMinusOne(int array[4][6]) **20 Marks**

The level is stored as an array of integers starting off with values 1 to 4 for. 0’s are not included.

Starting from 0,0 in the array test each column in the first row, then the second row , going from left to right until the end of the array is reached

Find 1 cluster of 3 or more identical items connected horizontally or vertically. Check for a horizontal line first and then a vertical line for each array position in turn.

Score the cluster by adding up the values (1 to 4) in the cluster.

Replace that cluster with ‘-1’s and returns a score

Example

Before

{ {0,0,3,1,3,4},

{0,0,2,3,4,3},

{0,0,1,3,3,2},

{0,0,2,2,2,2} };

After:

{ {0,0,3,1,3,4},

{0,0,2,3,4,3},

{0,0,1,3,3,2},

{0,0,-1,-1,-1,-1} };

Score would be 8

Return 8.

void PrintArray(int array[4][6]) **10 Marks**

Print the array in rows and columns. Only function you use cout.

FallDownAndReplace(int array[4][6]) **20 Marks**

‘Remove’ the -1’s in the array and the items above fall into their position. All items above fall into the ‘empty’ position below them.

New items are added to the top of the array with values of -2 and will not be part of the next cluster.

Example

Before :

{ {0,0,3,3,4,3},

{0,0,-1,3,4,3},

{0,0,-1,3,3,2},

{0,0,-1,2,3,3} };

After:

{{0,0,-2,3,4,3},

{0,0,-2,3,4,3},

{0,0,-2,3,3,2},

{0,0,3,2,3,3} };

**Testing**

Create tests that you did to show that the code is tested well. It is up to you to demonstrate that it works correctly.

# Part 2 : Parse C Code.

A.

void FindAllVariables(char variables[], char code[]) **15 Marks**

Write a function that takes 2 array of chars as parameters. The function fills the variable array to contain a comma separated list of the variable names that are declared in the code in the string. Only variables of type “int” and “char” need be included. Only one variable is declared per line.

Both variables array and code array have a max on 1000 chars.

Note strings are terminated by a 0.

For example

char code[1000] = “int main() \n{ \n int var1=10;\n int var2 =10;\n if(var1<10) \n {\n int var3=20;\n}\ n”;

char newline = ‘\n’;

char variables[1000] ;

FindAllVariables(variables ,code);

//variables now contains “var1,var2,var3”

B.

void FindVariablesInScope(char variables[], char code[], int lineNumber) **35 Marks**

Find variables in scope after a particular line number. The code won’t contain “for” loops. Only variables of type “int” and “char” need be included. Only one variable is declared per line. The only function declaration in the code is “int main()”.

Example:

char code[1000] = “int main() \n{ \n int numElephants=10;\n int var2 =10; \n if(var1<10) \n {\n int var3=20;\n}\nwhile (var2 <0) \n {\n int var4; \n} \n int var5 =10;\n ”;

char variables[1000] ;

FindVariablesInScope(variables,code,13); // the line with “int var5 = 10;”

//variables now contains “numElephants,var2,var5”

**Testing**

Create tests that you did to show that the code is tested well. It is up to you to demonstrate that it works correctly.