Assignment 4 - Island Perimeter

Due Feb 21 by 11:59pm **Points** 20 **Submitting** a file upload **File Types** h and cpp **Available** until Feb 24 at 12:01am

This assignment was locked Feb 24 at 12:01am.

ST. CHARLES COMMUNITY COLLEGE

CPT-182 - Programming in C++

Programming Assignment - Island Perimeter (20 Points)

(Number in Question Bank: Assignment **4.1**)

This question appeared in the job interview of **Apple**, **Amazon**, **Bloomberg**, **Facebook**, **Google**, and **Microsoft**.

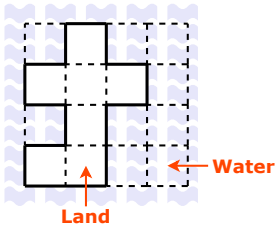
Program Overview

In this assignment, you will be given a map which is represented by a **2**-dimensional vector (vector of vectors). Each cell in the **2**-dimensional vector stores either **0** or **1** where **1** represents **land** and **0** represents **water**. A graphical explanation of the map is shown below:

2-Dimensional Vector

```
[[0, 1, 0, 0],
 [1, 1, 1, 0],
 [0, 1, 0, 0],
 [1, 1, 0, 0]]
```

Island Shape



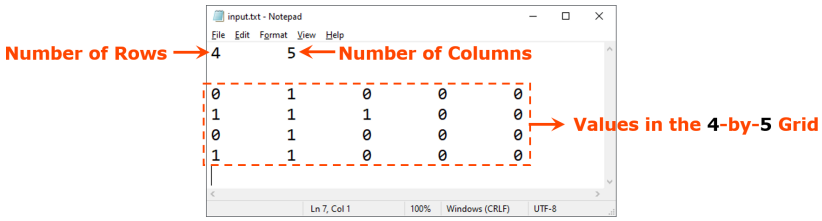
In this assignment, you can assume that all the following conditions are true:

- Grid cells are connected **horizontally** and **vertically** (**not** ~~diagonally~~).
- The grid is completely **surrounded by water**.
- There is exactly **1** island (**1** set of connected land cells).
- The island does **not** have ~~"lakes"~~ (water inside that is **not** connected to the water around the island).
- Each cell is a square with side length **1**.
- The grid is **rectangular**.

In this assignment, you are going to write a C++ program that reads in a **2**-dimensional vector (representing land/water) from an input file, **calculates the perimeter** of the island, and writes the result to an output file.

The Input File

- The input file is a **plain text file** (filename: `island.txt`).
- First row in the input file contains **2 positive integers**, which are the number of rows and number of columns in the grid (in that order).
- After the first row, values in the grid are listed in the input file (in **row-major** order).
- The picture below explains the data in the input file.



- Please note that you **cannot** assume (or guess) the dimensions of the grid. In other words, no matter how large the grid is, your program should correctly process it.

- Please refer to the **sample input files** to better understand the input file format.

The Output File

- The output file is a **plain text file** (filename: `perimeter.txt`).
- Your program writes the calculated **perimeter of the island** to the output file.
- Please refer to the **sample output files** to better understand the expected output file format.

The `calc_perimeter()` Function

- In this assignment, other than the **`main()`** function, you **must** write another function, **`calc_perimeter()`**.
- The **`calc_perimeter()`** function takes in a **2-dimensional vector of unsigned integers** (type `vector<vector<unsigned int>>`) as its **only** argument, which represents the grid of land/water. It is your responsibility to determine whether the argument should be passed by value, by reference, or by `const` reference.
- The **`calc_perimeter()`** function calculates the perimeter of the island in the grid, and returns the calculated perimeter.
- Docstring of the **`calc_perimeter()`** function is **required**, in which the function behavior, function parameter, and function return value are explained.
- For your convenience, in this assignment, please write the **`calc_perimeter()`** function and the **`main()`** function in the same file.
- The challenging part of this assignment is that you need to **set up a correct algorithm** to calculate the perimeter of the island. You need to use nested `for` loops to iterate through the **2**-dimensional vector. Hint: if the current cell has row index **`i`** and column index **`j`**, can your correctly find the row/column index of its possible **4** neighbors (top, right, bottom, left)?

The `main()` Program

- Your **`main()`** program should read the input file and get the dimensions of the grid first.
- Create a **2**-dimensional vector of unsigned integers with the dimensions just read from the input file.
- Read the rest values from the input file and store them in the **2**-dimensional vector just created.
- Call the **`calc_perimeter()`** function to calculate the perimeter of the island.
- Write the calculated perimeter to the output file.

Sample Input and Output Files (Click to Download)

Sample Input File 1 (https://drive.google.com/uc?export=download&id=1AcO2kMhnKsEDp0OU8neyZFkZikRXq_iF) Sample Input File 2 (
Sample Output File 1 (https://drive.google.com/uc?export=download&id=19IEtYH0YY1G-hn50j_imzK5Q_Sa66OsJ)Sample Output File 2

Assignment Submission and Grading (Please Read)

- Please upload all your **`.h`** (if any) and **`.cpp`** files (**not** the ~~entire Microsoft Visual Studio project folder~~) on Canvas.
- Before the assignment deadline, you can submit your work unlimited times. However, only your latest submission will be graded.
- At least **20**% of your code should be **comments**. All variable, function (if any), and class (if any) names should "make good sense". You should let the grader put **least effort** to understand your code. Grader will **take off points**, even if your program passes all test cases, if he/she has to put extra **unnecessary** effort to understand your code.
- Please **save a backup copy** of all your work in your computer hard drive.
- Your program will be graded (tested) using another valid input file (still named **`island.txt`**) to check whether it can generate the expected (correct) output file (with correct format and correct output values in it). As long as the input file is valid, your program should generate a correct output file. In other words, your program should work for **any** valid input file, **not** just the sample input files provided in the assignment instructions.
- In this class, you can assume that the input file (input data) is always **valid** and **has correct format**. You do **not** need to deal with ~~invalid input~~ or ~~error handling~~.
- Your work will be graded after the assignment deadline. All students will receive their assignment grades at (almost) the same time.