Assignment Example

Lines Example

The purpose of this example is to demonstrate an approach for solving a given problem with the requirements below for an actual user.

Below, there is a description of the scenario with some functional and technical requirements. However, it is up to you to choose how you are going to solve the problem and how the final technical solution of the application is going to look like.

1 Scenario:

https://en.wikipedia.org/wiki/Hydrothermal_vent

A customer needs a small utility application as an extension for a submarine control system for it to safely cross the ocean. Hydrothermal vents tend to form lines. Where two or more of these lines cross they constantly produce large, opaque clouds. For safety reasons these are to be avoided. Most of the hydrothermal lines are already known. They are provided via an input text file containing the coordinates of the start and end points of each line. The application should calculate - based on the hydrothermal lines of the input file - where the dangerous points for the submarine are located. A user interface (command line) shall allow to specify the input text file, which provides the basic data for the calculation. The file shall be analyzed, validated and parsed to retrieve valid hydrothermal lines for the calculation (details below). Further the application should calculate the crossing points of the hydrothermal venting lines and report it to the user on the command line and additionally write it into an output file.

1.1 Input File Parser:

- The input file parser should be able to efficiently read ANSI text files
- It shall only accept files, where the lines have the hydrothermal venting line format (defined below)
- If there is a line not following the hydrothermal venting line format, a proper error handling should happen
- If a line is identified, which is neither horizontal, vertical or diagonal (45°) the line should be skipped
- The parsing process should be responsive, abortable and have a progress bar
- The Input File Parser should be reusable

1.2 Hydrothermal venting line format:

Within the input file, the hydrothermal venting lines should be delivered in the following format:
X1,Y1 -> X2,Y2

X1 is the x coordinate of the starting location, Y1 is the y coordinate of the starting location, X2 is the x coordinate of the end location and Y2 is the y coordinate of the end location.

Where X1, Y1, X2 and Y2 are numbers. For simplicity reasons, all numbers are integers and only horizontal, vertical and diagonal lines (45° or a multiple of 45°) will be relevant for this example.

1.3 Hydrothermal lines:

A hydrothermal line consists of all integer coordinates between the start and the end point given.
Lines can be created like:

- 1,1 -> 1,3 which will cover points 1,1 1,2 and 1,3 (vertical)
- 9,7 -> 7,7 which will cover points 9,7 8,7 and 7,7 (horizontal)
- 1,1 -> 3,3 which will cover points 1,1 2,2 and 3,3 (diagonal)
- 9,7 -> 7,9 which will cover points 9,7 8,8 and 7,9 (diagonal)

1.4 Crossroad Calculation:

A crossroad is a coordinate, where at least 2 hydrothermal lines cross.

1.5 Hydrothermal Vent Crossroad format:

X and Y are integer values, which represent a point, where two or more hydrothermal vent lines have an intersection. N is an integer greater than 2, which represents the number of hydrothermal vent lines having the intersection on the point x and y.

All resources will expect this format to report crossroad points.

1.6 Output:

The customer wants to see the output on the console and so that he can hand the results over to another system of the submarine, he would like to define a path and a name, where an output file will be generated holding the same data as the console.

- In the first line he would like to have a header displaying the number of dangerous points
- After an empty line he would like to see all crossroads in the hydrothermal vent crossroad format

1.7 Additional information:

The solution must be written in C++17, Toolchain GCC or MSVC and handed in as a full source code together with project files and compiled files. A proper documentation must be provided how to build and run the application. The solution must run on Windows 10. The solution must be written by you. Please keep in that the quality of your solution reflects on your professional conduct.

When sending the source code, please delete the generated temporary and/or binary files from the solution to ensure that our mail system accepts the attachment. In case of any issues, please share the solution using some of the available platforms (e.g. GitHub, Google Drive, Dropbox, or similar)

1.8 Example:

User selects to read lines.txt as input file.
lines.txt:

```
0,9 -> 5,9
8,0 -> 0,8
9,4 -> 3,4
2,2 -> 2,1
7,0 -> 7,4
6,4 -> 2,0
0,9 -> 2,9
3,4 -> 1,4
0,0 -> 8,8
5,5 -> 8,2
5,5 -> 8,0
```

- File is valid as all lines have the proper format
- All lines except for the last are used for the calculation as thy are either horizontal, vertical or diagonal
- Last line is skipped as it is not horizontal, vertical or diagonal

With the example above, we should get the following diagram:

```
1.1....11.
.111...2..
..2.1.111.
...1.2.2..
.112313211
...1.2....
..1...1...
.1.....1..
1.......1.
222111....
```

He should be able to define an output file – in this case (same folder as the input file) he would like to call it crossroads.txt. After the calculation the console and crossroads.txt should show:

```
Number of dangerous points: 12


(0, 9) -> 2
(1, 9) -> 2
(2, 2) -> 2
(2, 9) -> 2
(3, 4) -> 2
(4, 4) -> 3
(5, 3) -> 2
(5, 5) -> 2
(6, 4) -> 3
(7, 1) -> 2
(7, 3) -> 2
(7, 4) -> 2
```