CSE 102 Programming Assignment 7

DUE

December 22, 2019, 23:55

Description

- This is an individual assignment. Please do not collaborate.
- If you think that this document does not clearly describes the assignment, ask questions before its too late.

Be careful with file names. You won't given a chance to correct any mistakes.

- Your program reads one text file:
  - `input.txt`
- Your program creates a text file:
  - `output.txt`
- Your program solves the following problem:
  - You are given a water level information `L`. Condition: `0<L<128`.
  - You are given a terrain information (elevation matrix `M`). Condition: $0 \leq$`M[i][j]`<128 $\forall$`i,j`<512.
  - Calculate how many `islands` are formed and return a coordinate value for each `island`.
  - Definition of an `island`: An island is the collection of terrain points(elevation values `M[i][j]`), above water(greater than `L` value), from which you can find a `path` to any other without having to pass the path through water.
  - Definition of a `path`: A path is an ordered set of points where successive points have either their `X` coordinates the same and their `Y` coordinates different only by one in magnitude (so can be `y - 1` or `y + 1`); or have their `Y` coordinates the same and have their `X` coordinates different only by one in magnitude (so can be `x - 1` or `x + 1`). But not both of them: in other words they have to touch either in the `X` direction or in the `Y` direction, and not in the diagonal.
  - If a piece of terrain is partially (or fully) located (touching) at the edge(s) of the `512 × 512` grid, still it will count as an `island`.

`input.txt`

- First line is a single integer. (The `L` value)
- The rest of the file has `262144` integers. Skip all the whitespace and read all of these integers and create a `512x512` matrix. Integers are listed in row-order.

`output.txt`

- The first line is an integer (number of islands)
- Each line after the first line holds coordinate information of a point on an island. (single point for each island)
- Example: Here there are `5` islands. (`#...` are comments, which are not printed.)

```
5
x_coordinate1 y_coordinate1                #coordinate of a point which is on island1
x_coordinate2 y_coordinate2                #coordinate of a point which is on island2
x_coordinate3 y_coordinate3                #coordinate of a point which is on island3
x_coordinate4 y_coordinate4                #coordinate of a point which is on island4
x_coordinate5 y_coordinate5                #coordinate of a point which is on island5
```

Remarks

- There will be at most `100` islands.
- You can allocate only one array.(create an array for the terrain data).

- You cannot declare any other array in your program.
- You cannot use dynamic allocation.
- Do not submit your code without testing it with several different scenarios.
- Write comments in your code.
- Do not print anything to `stdout` and `stderr`.
- Do not submit any of the files you used for testing.
- Do not submit your output file.

Turn in:

- Source code of a complete C program. Name of the file should be in this format: `<full_name>_<id>.c`.
- Example: `gokhan_kaya_000000.c`. Please do not use any Turkish special characters.
- You don't need to use an IDE for this assignment. Your code will be compiled and run in a command window.
- Your code will be compiled and tested on a Linux machine(Ubuntu). GCC will be used.
- Make sure you don't get compile errors when you issue this command : `gcc <full_name>_<id>.c`.
- A script will be used in order to check the correctness of your results. So, be careful not to violate the expected output format.
- Provide comments unless you are not interested in partial credit. (If I cannot easily understand your design, you may loose points.)
- You may not get full credit if your implementation contradicts with the statements in this document.

Late Submission

- Late submission is **NOT** accepted.

Grading (Tentative)

- `Max Grade : 100`.
- Multiple tests(at least `5`) will be performed.

All of the followings are possible deductions from `Max Grade`.

- `#define HARD_CODED_VALUES -10`.
- No submission: `-100`. (be consistent in doing this and your overall grade will converge to `N/A`) (To be specific: if you miss `3` assignments you'll get `N/A`)
- Compile errors: `-100`.
- Irrelevant code: `-100`.
- Major parts are missing: `-100`.
- Unnecessarily long code: `-30`.
- Using language elements and libraries which are not allowed: `-100`.
- Not caring about the structure and efficiency: `-30`. (avoid using hard-coded values, avoid hard-to-follow expressions, avoid code repetition, avoid unnecessary loops).
- Significant number of compiler warnings: `-10`.
- Not commented enough: `-5`. (Comments are in English).
- Source code encoding is not `UTF-8` and characters are not properly displayed: `-5`. (You can use 'Visual Studio Code', 'Sublime Text', 'Atom' etc... Check the character encoding of your text editor and set it to `UTF-8`).
- Missing or wrong output values: `Fails the test`.
- Output format is wrong: `-30`.
- Infinite loop: `Fails the test`.
- Segmentation fault: `Fails the test`.
- Fails 5 or more random tests: `-100`.
- Fails the test: `deduction up to 20`.
- Prints anything extra: `-30`.
- Unwanted chars and spaces in `output.txt`: `-30`.
- Submission includes files other than the expected: `-10`.
- Submission does not follow the file naming convention: `-10`.
- Sharing or inheriting code: `-200`.
- **IF YOU DON'T FOLLOW THE FILE NAMING CONVENTIONS YOU WILL GET `0.0`.**

Note: Some of these items are not independent. So, you cannot expect isolation of many of them. For example, if you cannot read input file correctly, you will fail to produce the correct output file. Partial grading is not guaranteed.