Lab 4 Skiing

Professor Patt likes skiing. When skiing, one would slide down from a higher place to a lower place to gain speed. But when one arrives at the bottom, he has to walk up or wait for others to pick him up. So every time Professor Patt always chooses the longest distance he can slide.

Suppose the skiing field is a rectangular area that is described by a two-dimensional array. Each element of the array denotes the height of the point.

Professor Patt can start from any point, and he can slide to one of the four adjacent points if its height is lower than the current height. He cannot leave the ski field when skiing.

We want you to write a program in LC-3 assembly language to help Professor Patt. Your program reads the map where Professor Patt is skiing and tells the longest distance he can slide.

Sample

Memory location x3200 stores N, the number of rows of that matrix. Memory location x3201 stores M, the number of columns of that matrix. The matrix is stored starting from x3202 and taking $N \times M$ following memory locations. For example, a 3x4 matrix as:

```
11  12  13  14
18  17  16  15
19  20  21  22
```

The matrix will be stored as:

Address	Value	Address	Value
x3200	3	x3207	17
x3201	4	x3208	16
x3202	11	x3209	15
x3203	12	x320A	19
x3204	13	x320B	20
x3205	14	x320C	21
x3206	18	x320D	22

After your program executing, the result should be stored in R2. For the example above, number x000C will be stored in R2.

Requirements & Notes

- Your program must use recursive ways to solve this problem.
- Your program should start at address x3000. You can read from the matrix, but cannot write.
- The starting point can be any position in the matrix, but your program is not required to tell where to start.
- When checking, the matrix will be no larger than 50 cells. That is, $N \times M \leq 50$. The time complexity would not affect your score.
- Please make your program less than 250 lines, and make it as readable as possible.

Grading

Lab 4 takes 10 points of the total score, consisting of the Check part (60%) and the Report part (40%).

Due: July 24, Friday

- **Check** (60% of each lab)
  - Find a TA to check your code in person, TAs may ask you questions when grading your lab assignment, you will get 100%, 80% or 60% of the checking score according to your response.
  - You can try again if you fail in checking, but there will be a penalty of -10% (of checking part) for each try.
- **Report** (40% of each lab)
  - English reports should be concise and carrying main ideas. Try to use the report to convince TAs that you complete the task by yourself.
  - Your lab report should contain the following contents:
    - **Introduction** to this lab. A brief explanation of what this lab requires your program do.
    - **Algorithm**. A brief explanation of your code. The complexity of your algorithm will not affect your score. You can use graphs to explain your algorithm.
    - Enough **Test Cases**. You can use the test cases in the Check part, or make your own cases.
    - **Discussion and Experience**. Write some thinkings about this lab.
    - **Source code** with sufficient comments. Comments begin with a semicolon ';'.
- **Penalty**
  - Delay: -10% per day. If more than 5 days, -100%.
  - Cheating: -100% of this lab. Besides, -10% of the total score.