Problem: You have a 2-dimensional array of integers. You may start at any location and can only move down or to the right and only if the value is smaller. You want to find the number of entries that make the longest path.

Input Format: Create **input3.txt** file in the same directory as the java and class files. "input3.txt" will be entered as a command line argument. The first line will be the value of the dimensions of the array separated by white space (rows columns). Each remaining line will be a whitespace separated list of the values of a row.

Output Format: The output will be the length (number of entries) of the longest path which starts on any square of the array, can only continue to a smaller numerical value, and can only continue to the adjacent square directly to the right or directly below. **You must find this value using dynamic programming (not brute force).**

Examples:

If input3.txt contains:

3 3

9 8 7

6 5 4

3 2 1

Then output will be:

5 → Because there is the path 9-8-7-4-1 with 5 squares and no path is longer.

If input3.txt contains:

4 6

19 11 16 19 14 30

12 13 14 15 16 31

14 28 26 28 22 32

16 50 16 14 10 33

Then output will be:

5 → Because there is the path 28-26-16-14-10 with 5 squares and no path is longer.

Note:

Your Java program should be commented, indented, and structured. Output should be sent to System.out. Please place all your files (.java, .class, .txt) in a directory named after you, zip them and submit them to canvas. Don't include any extra files and directories from IDE environment. The program must compile with the command javac *.java and run with the command java Project3 input3.txt. Remember input3.txt is a command line argument.

Don't place the classes in a package (use default package).