**10285 Longest Run on a Snowboard**

Michael likes snowboarding. That’s not very surprising, since snowboarding is really great. The bad thing is that in order to gain speed, the area must slide downwards. Another disadvantage is that when you’ve reached the bottom of the hill you have to walk up again or wait for the ski-lift. Michael would like to know how long the longest run in an area is. That area is given by a grid of numbers, defining the heights at those points. Look at this example:

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

One can slide down from one point to a connected other one if and only if the height decreases. One point is connected to another if it’s at left, at right, above or below it. In the sample map, a possible slide would be 24-17-16-1 (start at 24, end at 1). Of course if you would go 25-24-23-…-3-2-1, it would be a much longer run. In fact, it’s the longest possible.

**Input**

The first line contains the number of test cases *N*. Each test case starts with a line containing the name (it’s a single string), the number of rows *R* and the number of columns *C*. After that follow *R* lines with *C* numbers each, defining the heights. *R* and *C* won’t be bigger than 100, *N* not bigger than 15 and the heights are always in the range from 0 to 100.

**Output**

For each test case, print a line containing the name of the area, a colon, a space and the length of the longest run one can slide down in that area.

Sample Input

2

Feldberg 10 5

56 14 51 58 88

26 94 24 39 41

24 16 8 51 51

76 72 77 43 10

38 50 59 84 81

5 23 37 71 77

96 10 93 53 82

94 15 96 69 9

74 0 62 38 96

37 54 55 82 38

Spiral 5 5

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

**Sample Output**

Feldberg: 7

Spiral: 25

**Sample Input**

**2**

1. 輸入資料筆數

最初決定資料數目且接連輸入完畢後連續輸出。

例如f(x)=x+1

Input

2

1

3

output

2

4

#include <iostream>

using namespace std;

void testFunction(int inputNum)

{

cout<<inputNum+1<<endl;

}

int main()

{

int dataNum;

int inputNum[100];

cin>>dataNum;

if(dataNum>0)

{

for(int i=0;i<dataNum;i++)

{

cin>>inputNum[i];

}

cout<<"output:"<<endl;

for(int i=0;i<dataNum;i++)

{

testFunction(inputNum[i]);

}

}

else

return 0;

return 0;

}

Spiral 5 5

1. 輸入資料名稱及大小

名稱存入string array即可

矩陣大小部分較為困難

先前嘗試過類似

Cin>>www;

Int intArr[www];

先輸入整數大小在做陣列宣告的做法但行不通。

從另一個觀點看，假設輸入不會錯誤，那麼將n\*m的資料輸入直接看為m行，每一行的資料再以string.split(“ “)分隔開來。

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

1. 資料內容