Springer Sun Spot Analyser Test

Introduction

NASA wishes to analyse areas of the Sun for thermal activity. Their sensors provide raw data which can visualised as a *5* by *5* grid of raw heat measurements that represents the Sun's surface, i.e.

	0	1	2	3	4
0	5	3	1	2	0
1	4	1	1	3	2
2	2	3	2	4	3
3	0	2	3	3	2
4	1	0	2	4	3

The raw data for the analysis comprises of a list of integer measurements within the range from 0 (low heat) to 5 (high heat); therefore the raw data for the above grid would be as so:

```
5 3 1 2 0 4 1 1 3 2 2 3 2 4 3 0 2 3 3 2 1 0 2 4 3
```

This data needs to be analysed to find likely areas of increased solar activity. The analysis consists of generating an *Solar Activity Score* for each location on the grid. The score is determined by adding the location's own raw heat value to its adjacent raw heat values. For instance, the score of location (1,1) above would be as follows:

```
5 + 3 + 1 +

4 + 1 + 1 +

2 + 3 + 2

= 22
```

When dealing with locations around the edge of the grid the score should ignore values outside the grid. For instance the score of location (0,0) is as follows:

```
5 + 3 +

4 + 1

= 13
```

Problem 1

Given that the raw data comprises of the following input:

```
5 3 1 2 0 4 1 1 3 2 2 3 2 4 3 0 2 3 3 2 1 0 2 4 3
```

Please provide a program that parses this input and outputs the scores for each grid location in the same order. The expected result is:

```
13 15 11 9 7 18 22 20 18 14 12 18 22 23 17 8 15 23 26 19 3 8 14 17 12
```

You can use any mechanism you wish to provide input into your program but you must provide evidence that the program's output meets the requirements.

Problem 2

A new requirement has been added so that variable size grids can be processed. To cope with this, an additional input will be added at the beginning of the raw data which signifies the dimensions of the grid.

n	Grid
3	4 2 3 2 2 1 3 2 1

In the example above, *n* is the size of the grid and *grid* is a list of numbers that form the grid, starting with row 0.

List of numbers above represents a request for scores for a 3x3 grid that looks like so:

4	2	3
2	2	1
3	2	1

Please provide an updated program that meets this new requirement. Please also provide a unit test to prove the additional functionality using the following test data:

Input:

```
3 4 2 3 2 2 1 3 2 1
```

Expected output:

```
10 14 8 15 20 11 9 11 6
```

Problem 3

Another new requirement has now been added for your program. Rather than return a complete list of all *solar activity scores* it is now important to return only the highest relevant scores.

To meet this need, an extra value will be added to the beginning of the input:

t	n	Grid
1	3	4 2 3 2 2 1 3 2 1

The required output of the analysis is to provide an ordered list of locations containing the *t* number of highest rated locations and their respective scores.

Output should be a list of locations and their scores in the following form:

`(x,y score:score)`

i.e. the result that should be returned for the above input is:

`(1,1 score:20)`

Please provide proof that your updated program now fulfills this need for the following two test cases:

Test 1

Input:

`1 5 5 3 1 2 0 4 1 1 3 2 2 3 2 4 3 0 2 3 3 2 1 0 2 4 3`

Expected output:

`(3,3 score:26)`

Test 2

Input:

`3 4 2 3 2 1 4 4 2 0 3 4 1 1 2 3 4 4`

Expected output:

`(1,2 score:27)(1,1 score:25)(2,2 score: 23)`