South African Computer Olympiad
Online Camp 2008
Day 1

Overview

Author	**Carl Hultquist**	**Carl Hultquist**	**Bruce Merry**
Problem	**walls**	**area**	**change**
Source	walls. walls. walls. walls. walls. walls.	area. area. area. area. area. area.	change. change. change. change. change. change.
Input file	walls.in	area.in	change.in
Output file	walls.out	area.out	change.out
Time limit	5 seconds	10 seconds	2 seconds
Number of tests	10	10	10
Points per test	10	10	10
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Blow-torching Walls

Author

Carl Hultquist

Introduction

After Farmer John's cows tried to escape the farm recently, Farmer John decided that he has had enough: these cows must be sent to the abbatoir! The cows, who have been spying on Farmer John, have discovered this... and are very worried. They **really** need to escape now, otherwise they'll end up as roast beef on Farmer John's dinner table!

There is only one chance of escaping: and that is through the underground maze that runs beneath the farm. The cows raided Farmer John's office during the night and found a map of the maze. The only problem is that Farmer John has blocked up many parts of the maze with concrete wall... But luckily for the cows, they have also raided his toolshed and have stolen a blow-torch!

Blow-torching an entire piece of wall takes a **long** time: so the cows need to work out the smallest number of walls that they'll have to blow-torch away in order to escape the maze, and also find the shortest route that needs this number of walls blow-torched away.

Task

Your task is to help the cows plan their escape route. The maze consists of a number of square areas which are either open or which are a wall. The cows can move from any one of these square areas to any of the adjacent square areas (first blow-torching any wall that might be in the way). Moving from one square to another counts as moving a distance of 1 unit. You must:

1. Determine the smallest number of walls that need to be blow-torched for the cows to travel from the given starting point in the maze to the given ending point.
2. Determine the shortest route from the starting point to the ending point that requires Z walls to be blow-torched, where Z is the number determined in part 1. There may be more than one such route: you only need to determine one of them.

Example

Suppose the maze looks like the diagram below, where a '.' indicates an open part of the maze, and a 'W' indicates a wall, and that the cows need to get from the top-left hand corner to the bottom-right hand corner.

```
.W...
...WW
.W.W.
.WWW.
.WW..
```

The smallest number of walls that the cows need to blow-torch is 1, and they can get through the maze most quickly by moving: down, right, right, down, right (blow-torching this wall), right, down and down, which is a total of 8 units of movement.

Input (walls.in)

The first line of input will contain two integers, W and H, which indicate the width and height of the maze respectively. The second line of input will contain two integers, SX and SY, indicating the X and Y co-ordinates respectively of where the cows enter the maze. The third line of input will contain two integers, EX and EY, indicating the X and Y co-ordinates respectively of where the cows must exit the maze. The next H lines will each contain W integers, separated by spaces, that describe the maze. Each of these integers will either be 0 or 1: 0 indicates that there is no wall at that point in the maze, 1 indicates that there **is** a wall.

The upper-left co-ordinate of the maze is $(1, 1)$ and the lower-right co-ordinate is (W, H). This is illustrated below:

```
(1,1)..........
...............
..........(W,H)
```

Sample input

```
5 5
1 1
5 5
0 1 0 0 0
0 0 0 1 1
0 1 0 1 0
0 1 1 1 0
0 1 1 0 0
```

Output (walls.out)

The first line of output must contain two integers, Z and L. Z is the smallest number of walls that need to be blow-torched (determined in part 1 above), and L is the length of the shortest path that requires Z walls to be blow-torched. The next $L+1$ lines should each contain two integers, X and Y, which indicate the co-ordinates of the route that the cows must take.

Sample output

```
1 8
1 1
1 2
2 2
3 2
3 3
4 3
5 3
5 4
5 5
```

Constraints

- $1 \leq W, H \leq 1000$
- The starting location, (SX, SY), will never have a wall on it.

50% constraints

- $1 \leq W, H \leq 250$

Time limit

5 seconds.

Scoring

If your output for a test-case differs in any way from the format prescribed above, then your program will score 0% for that test-case. If your path fails to lead the cows from the starting point to the ending point, then you will score 0%. If the number of walls you need to blow-torch in the path is different from Z, then you will score 0%.

Otherwise, let Q be the optimal number of walls that needed to be blow-torched. Let M be the optimal route length for a route requiring Q walls to be blow-torched. Put $K = \max(M, L)$. Your score is then given by $S = [100 - 30 \times (Z - Q)] \times (\frac{M}{K})^{Z-Q+1}\%$.

Thus if you find the optimal number of walls, and the optimal route for that number of walls, your solution scores 100% :-)

Introduction

Jack desperately wants to marry his school sweetheart, Jill. But before Jill will consent to marriage, she sets a hard puzzle for Jack so that he can show how serious he is about marriage.

Jill takes all the rectangular love letters that Jack has ever sent her, and scatters them on the floor, with some letters overlapping others. For Jack to pass the test, he needs to work out what area of the floor is covered by letters.

Task

Given the positions of all the rectangular letters on the floor, find the total area of floor covered. Note that letters may overlap, meaning that some areas of the floor might be covered by more than one letter.

Example

Given four letters with the following sets of top-left and bottom-right co-ordinates in (x; y) format:

- $(1; 2)$ to $(3; 4)$
- $(1; 1)$ to $(2; 2)$
- $(0; 6)$ to $(7; 8)$
- $(2; 3)$ to $(5; 7)$

The total area of the floor covered by letters is 27 because the sum of the areas of the letters is 31, but 4 of these units are comprised of overlapping letters.

Input (area.in)

The first line of area.in will contain a single integer, N, which specifies the number of rectangles that Jill has thrown onto the plane. The next N lines will each contain 4 integers, x_1, y_1, x_2 and y_2, which specify the bottom left and top right corners of each rectangle. So $(x_1; y_1)$ specifies the bottom left corner and $(x_2; y_2)$ specifies the top right corner.

Sample input

```
4
1 2 3 4
1 1 2 2
0 6 7 8
2 3 5 7
```

Output (area.out)

area.out should contain a single integer which specifies the area occupied by the N rectangles in the input.

Sample output

```
27
```

Constraints

- $0 \leq xi, yi \leq 10000$
- $2 \leq N \leq 10000$

50% constraints

- $0 \leq xi, yi \leq 500$
- $2 \leq N \leq 500$

Time limit

10 seconds.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

Making Change

Author

Bruce Merry

Introduction

The cows have decided that they are tired of the strange American system of coins, being particularly annoyed by the large gap between the quarter ($0.25) and the $1 coins that force one to carry lots of quarters. They are going to create their own system of coins with sensible values.

However one feature of the American system that they like is that one can always make up a value with a "greedy" algorithm and get the minimal number of coins. The greedy algorithm says that one should always pick the largest value coin that is not greater than the amount still needed. So for example to make 83c in the US system (which has coins of values 1c, 2c, 5c, 10c, 25c and $1), one first picks three 25c coins to make 75c, then a 5c to make 80c, then a 2c and finally a 1c. This results in six coins, and there is no way to make 83c with fewer coins. However if the coins had been 1c, 4c and 6c, then making 8c using a greedy algorithm takes 3 coins (6+1+1) even though it is possible to make 8c using 2 coins (4+4).

Task

You must write a program to determine whether the cow's proposed set of coins satisfies the greedy property, and if not compute the smallest value for which the greedy algorithm is not optimal.

Input (change.in)

The first line is N, the number of coins. Each of the next N lines contains the value of a coin, in cents. Values may be repeated (e.g. for special editions). It is guaranteed that at least one coin will have the value of 1c.

Sample input

```
3
1
6
4
```

Output (change.out)

If the coin set does not have the greedy property, output the smallest value for which the greedy algorithm is non-optimal, followed by the minimum number of coins required to create this value (space separated).

If the coin set has the greedy property, output "−1 −1".

Sample output

```
8 2
```

Constraints

- $1 \leq N \leq 100$
- $1 \leq$ each coin ≤ 10000

50% constraints

- $1 \leq N \leq 10$

Time limit

2 seconds.

Scoring

An incorrectly formatted output file scores 0.

If the coin set has the greedy property, a correct answer scores 100% and an incorrect answer scores 0.

If the coin set does not have the greedy property, then 50% is awarded for getting the first output correct. If (and only if) this output is correct, 50% is awarded for getting the second field correct.