Old Mutual - CSSA Computer Olympiad DAY 2

Overview

Problem	Stars	Sticks	Bit Codes
Program name	`stars.exe`	`sticks.exe`	`codes.exe`
Source name	`stars.pas` `stars.cpp` `stars.java`	`sticks.pas` `sticks.cpp` `stars.java`	`codes.pas` `codes.cpp` `stars.java`
Input file	`stars.in`	`sticks.in`	`codes.in`
Output file	`stars.out`	`sticks.out`	`codes.out`
Time limit per test	5 seconds	1 second	10 seconds
Number of tests	10	10	10
Points per test	10	10	10
Total points	**100**	**100**	**100**

The maximum total score for Day2 is 300 points.

Stars

Author: David Turner

Description

Jack and Jill are building a spaceship in which they will visit the galaxy. Being particularly sensible ten-year-olds, they wish to calculate precisely how much antimatter needs to be stored in the warp core.

Jack and Jill's journey will consist of several legs (trips between stars). On their return they will have visited each star in the galaxy. It is possible that they will visit a particular star more than once, and they won't necessarily take the shortest route. They can start at any star and will complete their journey when they return to that star.

The warp core can be recharged at any star, so what they need to know is the *maximum* amount of fuel that will be used in any leg of their journey around the galaxy. Since antimatter containment fields are expensive, the smaller they can make the fuel tank, the better. Their journey will be planned so as to minimize the size of the fuel tank.

You've decided to help them by working out the length of the longest leg of their journey. The location of every star is given in galactic coordinates (x, y, z).

The distance between a pair of stars can be calculated by:

$$dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Task

If the length of the longest leg of a journey P around the galaxy is *L(P),* your task is to find the minimum *L(P)* in the set of *all* possible journeys that visit *all* the stars in the galaxy.
In order that Jack and Jill can calculate the smallest tank they need, you must report the numbers of the two stars between which you find the minimum *L(P).* The stars are numbered in the order that they appear in the file, 1 to *N*, where *N* is the number of stars in the galaxy.

Input (stars.in)

The first line of input file "stars.in" is a single integer *N*, the number of stars in the galaxy. *N* lines follow this, each containing 3 integers, the galactic coordinates of the stars.

Sample input:

stars.in

```
4
0 0 0
1 0 0
0 1 0
-5 -2 0
```

Output (stars.out)

Your output will be the pair of stars between which the longest journey is made. The output consists of two lines each with a single integer, the number of a star.

Sample output:

stars.out

```
1
4
```

Constraints

$2 \bullet N \bullet 10000$
$-2^{13} \bullet x, y, z < 2^{13}$

Time Limit:

Maximum time allowed per test case is 5 seconds

Scoring:

For each data set the score is calculated as:

$$10 \times \frac{\text{shortest\_distance}}{\text{output\_distance}}$$

Decimal scores are rounded down (floor) to the nearest integer, this means 10/10 for a perfect solution.
0 points if output_distance is too short.

Sticks

Author: Graham Poulter

Description

Jill has arranged a large number of vertical and horizontal sticks on a plane and challenged Jack to find some properties of the pattern of sticks, including the number of clumps that they form. Fortunately for Jack, you are here to help him. The sample test case is depicted below:

Two sticks are counted as intersecting if they share at least one block. The block with "468" in it represents three intersections: 4-6, 4-8, and 6-8. Each intersection must only be counted once, so don' t count the "46" block as another intersection.

If you glued the sticks together wherever they intersected, then the clumps would be those bunches of sticks which could be picked up as single pieces. A single stick not intersecting any others also counts as a clump.

8		8							
7		8							
6		8			5				
5		468	46	4	45	4	47	7	7
4		8			5				
3		8		3	5				
2				3					
1				3		1			
0		2	2	23	2	21			
	0	1	2	3	4	5	6	7	8

Task

A) Find the number of intersections in the set of sticks.
B) Find out how many clumps the sticks form.
C) Find the number of sticks in the largest clump.

Input ("sticks.in")

Line 1: One integer, N, the number of sticks
Lines 2..N+1: Four integers, X0, Y0, X1 and Y1 representing the end-points of the sticks. Vertical sticks have X0 = X1 and horizontal sticks have Y0 = Y1.

Sample Input

sticks.in

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

Output ("sticks.out")

Line 1: One integer, the number of intersections.
Line 2: One integer, the number of clumps.
Line 3: One integer, the number of sticks in the largest clump.

Sample Output

sticks.out

```
7
2
5
```

Constraints

1 • N • 30000
0 • X0, Y0, X1, Y1 • 10000
All sticks have a length of at least 2 blocks

Time Limit

Maximum time allowed per test case is 1 second.

Scoring

There will be 10 test cases
maximum of 10 points per test case:

3 points for the right number of intersections
5 points for the right number of clumps
2 points for the right number of sticks in the largest clump

Bit Codes

Author: Carl Hultquist

Description

Jack and Jill are on the hunt to find the fabled "Area 51" laboratories. They manage to sneak into the perimeter of Area 51, and soon have found their way to a back entrance while the security camera is looking in a different direction. The door is, however, guarded by a state-of-the-art "Bit Code (tm)" lock which is supposedly impossible to get past without knowing the key.

1	1	0	0	1
0	1	0	1	1
1	2			

The lock looks something like the picture above, with two rows of 1' s and 0' s. Below the lock is a row of buttons, each with a number on it. Pressing a button causes some of the digits in the top row to interchange. As soon as the top row is the same as the bottom row, the lock opens.

Task

You need to help Jack and Jill work out which buttons to press in order to open the lock. But you need to find the best way of doing it before the security camera turns around and busts them!

Input (codes.in)

The first line of "codes.in" will contain two integers N and M, 0 < N • 23 and 0 < M •25. The next line will contain a string, called S. The line after that will also contain such a string, called E. Both E and S will consist of exactly N characters, and these characters can only be ' 1' or ' 0' . S is the top row in the "Bit Lock (tm)", and E is the bottom row.

The next M lines represent the buttons on the "Bit Lock (tm)" and have the following format:
Each line will start with a single integer L that will be followed by exactly L integers K_1 to K_L. Each K_i will be such that 0 < K_i • N, and the K_i' s represent the bits in S that are affected by pushing this button. Specifically, if this button is pushed, then the bits in S rotate as follows: $K_1 \rightarrow K_2 \rightarrow K_3 \rightarrow ... \rightarrow K_n \rightarrow K_1$. 0 < L • N, and no K_i will be repeated in a button.

codes.in

```
5 2
11001
01011
3 1 2 3
3 3 4 5
```

Output (codes.out)

The first line of "codes.out" must contain a single integer T, representing the smallest number of button presses needed to change S into E. The next T lines must each consist of a single integer representing the button pushed.

codes.out

```
3
1
2
2
```

Consider this step by step:

After pressing button 1, the **1** in bit 1 moves to bit 2, the **1** in bit 2 moves to bit 3, and the **0** in bit 3 moves to bit 1 changing S to **01101**. Now pressing button 2 moves the **1** in bit 3 to bit 4, the **0** in bit 4 to bit 5, and the **1** in bit 5 to bit 3, giving **01110**. Pressing it again performs the same rotation, giving the result we need.

Constraints

0 < N • 23
0 < M • 25
It is guaranteed that there is a way of changing S into E.

Time Limit:

Maximum time allowed per test case is 10 seconds

Scoring:

If your output file differs from the format above, you score 0. If the button-presses you specify do not change S into E, you score 0. Otherwise, your score is calculated based on how close your answer T was to the optimal answer O. This is given by:

$$X = 10 - \frac{T}{O} * \ln \frac{T}{O}$$

If X < 0, you score 0.
If X is not an integer, it is rounded down to the nearest integer to give your score.