ECOO 2015

Programming Contest Questions

Provincial Finals

May 9, 2015

Sheridan | Get Creative

Questions made possible in part through the support of the Sheridan College Faculty of Applied Science and Technology.

Problem 1: Just the Factoradics, Ma'am

The factoradic number system is a number system that uses factorials as the place value for each digit. In case you don't know, the factorial value of an integer n is written "n!" and it evaluates to what you get when you multiply together all the integers from 1 to n, like this:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

In the factoradic number system, the rightmost 'digit' has a base value of 0!, the second digit has a base value of 1!, the third digit has a base value of 2!… and the nth digit has a base value of (n-1)! Each digit of a factoradic number must be less than or equal to n, where n! is the place value of the digit.

For example the factoradic number $3\ 0\ 1\ 0_!$ has 4 digits. See the table at right for a breakdown of place values and highest digit values. To convert a number from the factoradic number system to base 10, you multiply each digit by its base value and sum them all up.

3	**0**	**1**	**0**	
3!	2!	1!	0!	Place value
3	2	1	0	Highest digit

$$3\ 0\ 1\ 0_! = 3 \times 3! + 0 \times 2! + 1 \times 1! + 0 \times 0! = 3 * 6 + 1 * 1 = 19$$

Because larger factoradic numbers have digits larger than 9, we separate each digit with a space, as shown in the example below:

$$11\ 9\ 10\ 0\ 8\ 5\ 3\ 2\ 0\ 1\ 1\ 1\ 0_!$$

DATA11.txt (DATA12.txt for the second try) will contain 10 test cases. Each test case consists of a single line containing a set of non-zero decimal digits concatenated together. The maximum number of digits is 30. Each line represents a single factoradic number which is missing all its 0s and all the spaces that are supposed to separate the factoradic digits. Find the smallest legal factoradic number that can be made using the decimal digits in their current order. You may add as many zeroes as you wish and place them anywhere you wish. Output your answer as a factoradic number with each factoradic digit separated by a space.

Sample Input

```
72
11
12111722
11111722
1566111414121811722
```

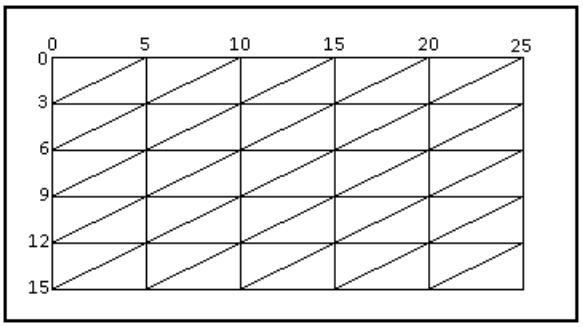
Sample Output

```
7 0 0 0 0 2 0 0
1 1 0
1 2 1 1 1 7 0 0 0 2 2 0 0
11 1 1 1 7 0 0 0 2 2 0 0
15 6 6 11 14 1 4 12 1 8 1 1 7 0 0 0 2 2 0 0
```

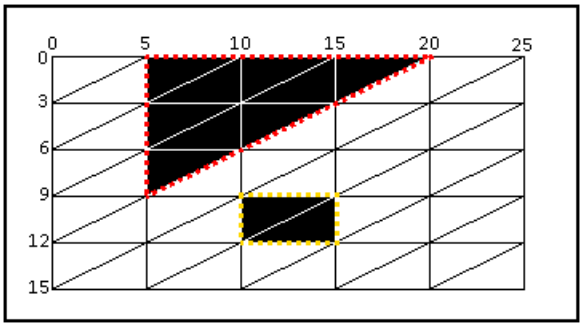
Problem 2: Tiling Trouble

A grid is subdivided into right-angled triangles with an arbitrary base (horizontal) and height (vertical). In the images below, the base of each triangle is 5 and the height is 3. Each triangular tile is made of a special color-changing material. By “reversing the polarity” of a tile you can trigger a color change.

Given a starting point (the X and Y coordinate of the right angle of a tile) and measurements in the horizontal and vertical directions, larger shapes can be specified. You can make right-angled triangles and rectangles in this way.



The diagram below shows a right-angled triangle at point (5,0) with base and height of 3 tiles, and a rectangle at point (10,9) with a base and a height of 1 tile. Note the shaded tiles that are completely covered by the shapes.



If we reverse the polarity on those tiles we will change the color.

As we continue to define shapes we will continue to reverse the polarity on all tiles that lie completely within the shapes, and we will change their colors.

DATA21.txt (DATA22.txt for the second try) will contain 10 test cases. The first line of each test case consists of two integers B and H ($1 \leq B, H \leq 1000$) that specify the base and height of the right-angled triangle tiles on the plane. B lies on the X axis while H lies on the Y axis.

The second line of the test case is the sequence of colors (e.g.m “Red”, “Blue”, “SunshineYellow”, etc.). The sequence will contain between 1 and 10 colors. Each color is a single word and the colors are each separated by a single space. Each tile starts with the first color and changes to the next color in the sequence each time its polarity is reversed. If this happens enough times, the color sequence will restart from the beginning. Each color in the list will be different from the color immediately preceding it. Tiles share their edges, so reversing the polarity of either tile along an edge changes the color of the points on the edge.

The third line of each test case is an integer N ($1 \leq N \leq 5000$), that specifies the number of shapes you will use for reversing the polarity of the tiles. The next N lines will each consist of a shape described in one of the two following ways:

> T X Y Tb = Triangle with the right angle at point (X, Y) and a base of Tb tiles, where $0 \leq X, Y \leq 8000$ and $-30 \leq Tb \leq 30$
>
> R X Y W H = Rectangle with a corner at point (X, Y) , a width of W tiles and a height of H tiles, where $0 \leq X, Y \leq 8000$ and $-30 \leq W, H \leq 30$

The last 500 lines of a test case will each consist of two integers Px and Py ($0 \leq Px, Py \leq 38000$) representing a point on the grid.

For each unique color in the sequence, you must output the number of points that will end up that color after the appropriate tiles have had their polarities reversed once for each of the shapes in the list. The colors and counts should be printed on a single line using the exact format shown below. The order in which you output the colors on each line is not important.

Note that in the sample input, there are 3 test cases, each with 5 points but the real data will have 10 test cases with 500 points each.

Sample Input

```
1 3
White Black
10
R 78 66 19 2
T 18 96 8
T 2 78 1
R 87 78 66 2
R 33 45 55 11
R 50 36 8 3
T 59 72 40
R 19 15 67 6
T 51 54 17
R 98 54 16 28
123 60
33 64
196 177
99 145
88 162
3 3
White Black
10
R 45 63 8 23
R 42 18 19 21
R 45 42 28 30
R 18 48 19 12
R 3 63 22 4
T 75 72 28
T 84 72 22
T 42 0 22
T 78 57 29
R 24 36 25 6
108 44
24 164
177 40
122 82
115 37
3 2
White Black
10
T 75 44 27
R 12 36 6 26
T 21 12 4
T 3 38 22
T 45 40 17
T 75 32 30
T 33 80 1
R 6 42 19 49
T 15 92 24
R 72 66 5 16
172 120
195 82
61 99
140 169
132 155
```

Sample Output

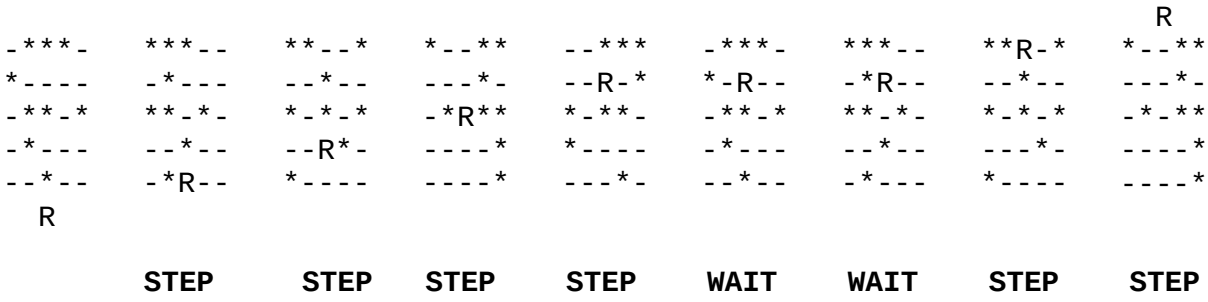
```
Black:1 White:4
Black:1 White:4
Black:0 White:5
```

Problem 3: Roadie Cross

Somewhere in a parallel world, one of Taylor Swift's roadies is trying to get across a busy road carrying a huge load of equipment. She's so loaded down that she can't go left, right or backwards. She can only step forward or wait where she is. Can you help her get across the road?

Roads are strange in this parallel world. Each road consists of a number of lanes of traffic, but the first lane moves left, the next moves right, and then the lanes continue to alternate directions until you get to the other side. Each lane contains a looping pattern of cars and empty car-sized areas.

Time and space are also strange. Time moves forward in discrete chunks and people and cars can jump from one location to another in car-sized or lane-sized jumps without having to cross the intervening space. As a result, a road crossing situation for the roadie described above could unfold as shown in the series of pictures below (R is the Roadie, asterisks are cars).



DATA31.txt (DATA32.txt for the second try) will contain 10 test cases. The first line of each test case contains two integers L and W separated by a space. L indicates the number of lanes L ($1 \le L \le 100$) and W indicates the width of each lane ($W = 2k+1$, $0 \le k \le 250$). The next L lines contain a representation of each lane in which a hyphen character (ASCII 45) indicates an empty area and an asterisk (ASCII 42) indicates a car. Every lane contains at least one car. The roadie starts her journey below the bottom lane of the road and centered with respect to the initial configuration of traffic patterns on the road (as in the example above). In each time step she can move forwards (up) or wait. Your job is to output the minimum number of time steps for the roadie to cross the road. If it is not possible for her to cross the road at all, you should output the string "Not Possible" with the first letter of each word capitalized and the rest lower case.

Sample Input

```
5 5
-***-
*----
-**-*
-*---
--*--
```

```
4 13
-*----*--*---
--*----------
**--*---*****
----***-*****
```

```
4 13
*------*****-
---****---*--
--**--*-****-
***-**--*--*-
```

Sample Output

```
8
Not Possible
5
```

Problem 4: Under the Rug

You're a member of the custodial staff at a large convention centre. You came in for your shift this morning to find that the contractors painting the ceiling overnight had an accident and now there's a huge paint stain on the floor. It's already dried. The boss wants you to do something about it.

Fortunately, the convention centre has a huge stock of square carpet tiles in a range of sizes that you can put down on the floor to cover the stain. These carpet tiles are so thin that you can overlap them as much as you want without creating a tripping hazard, but you know from past experience that the boss is really picky about how you place them.

The floor of the convention centre is divided into a grid such that each square in the grid is 1m x 1m. The boss likes the carpet tiles to be placed so that each grid square is either totally covered or totally uncovered. The boss will not like it if you put carpet tiles over any grid square that does not have at least a little bit of paint on it. The boss will also not like it if you try to place a rug that doesn't fit within the boundaries of the convention centre floor.

DATA41.txt (DATA42.txt for the second try) will contain 10 test cases. Each test case will consist of N lines of N characters each ($1 \le N \le 200$), representing the grid squares of the convention centre floor. Any grid square with paint on it is marked with an asterisk character (ASCII code 42). Grid squares with no paint on them are indicated with a hyphen character (ASCII code 45).

For this question, you will send your output to a file named "*solution.txt*". You will be allowed 60 seconds of execution time instead of the usual 30 seconds.

Your job is to output a single integer M for each test case (M>0) to indicate the minimum number of rugs required to cover the spill without covering any grid squares that are not part of the spill. This should be followed by M lines, each representing one of the M rugs in your solution. Each rug is represented by three integers T, L and S representing its top row, leftmost column and size. The coordinate system starts at (0, 0) and increases downwards for rows and to the right for columns. You can assume that you have an unlimited supply of SxS rugs of all sizes ($1 \le S \le 200$, in metres).

Note that the sample input below only contains 2 test cases, but the real data files will contain 10.

Sample Input

```
-----
--***
-***-
-**--
-----
----------
--**------
--****----
--******--
--******--
---*****--
---*****--
-----**---
-----**---
------*---
```

Sample Output

```
3
2 1 2
1 2 2
1 4 1
7
3 4 4
2 2 3
7 5 2
3 3 4
1 2 2
2 3 3
9 6 1
```

Notes on Grading

When you run your program, you will be asked to move your *solution.txt* file onto a USB memory drive supplied by the judges. Your output will be scored by a judging app which will assign a base score of 5 points for each solution that is valid (all the squares with paint on them are covered and no squares without paint are covered) and then up to 5 more points based on how close the size of your solution was to the target size. The judges can show you the target sizes after they have collected your output.

Be sure that your data file matches the required output format exactly. The judging app might be able to recover from badly formatted or invalid data, or it might not. If your data is not formatted exactly correctly, you might get some points or you might get nothing.

Question Development Team
Sam Scott (Sheridan College)
Kevin Forest (Sheridan College)
Stella Lau (University of Cambridge)
Dino Baron (University of Waterloo)
Greg Reid (St. Francis Xavier Secondary School, Mississauga)
John Ketelaars (ECOO-CS Communications)
David Stermole (ECOO-CS President)