



Preliminary Electronic Contest

11th BME International 24-hour Programming Contest

<http://ch24.org>



Preliminary Electronic Contest

Welcome to the testing round of the 11th BME International 24-hour Programming Contest!

This document is the problem set for the Preliminary Electronic Contest to be held on February 12th, 2011.

The PreEC provides a way for teams to familiarize themselves with our submission system and the general atmosphere of the competition. Whether teams participate in this testing round or not, or whatever results they achieve, will not have any consequences later in the competition.

The four problems we selected for the PreEC may not be the kind of problems you would consider especially challenging, however they provide some clues about the Electronic Contest - the required tools, the usage of our submission system and so on.

Rules

The Preliminary Electronic Contest contains four problems. You have all the time in the world to solve them, but we take submissions from 9:00 to 21:00 CET. The inputs of the problems can be found in a zip file that you have probably already downloaded from the website. Each problem will have exactly 10 test cases.

You can use any platform or programming language to solve the problems. We are interested only in the output files, you don't need to upload the source code of the programs that solved them. Once you are done, you can upload your output files via the submission site: <http://ch24.org/sub>. Your solutions will be evaluated on-line.

There are two major problem types:

- Non-scaled problems: problems that have an exact solution. When submissions to these are evaluated, a final score is given immediately. From one team, only one correct submission will be accepted for each input (since the input is either solved or not). In the PreEC, P, R, and S are non-scaled problems.
- Scaled problems: problems that do not have a known "best" solution. Outputs for these problems compete against each other, and scores are scaled according to the best uploaded output. A team may submit multiple correct submissions to one input (only the latest submission will be taken into consideration). In the PreEC, only Q is a scaled problem.

Note that points are awarded per output file and not per problem. If your solution only works for some of the input files, you will still be awarded points for the correct output files. A single output file however is either correct or wrong - partially correct output files are not worth any points.

Additional information for non-scaled problems:

Be quick about uploading the output files, because the scores awarded for every output file decrease with time. Uploading it just before the end of the contest is worth **70%** of the maximum points achievable for the test case. During the contest its value decreases linearly with time. However you should also be careful with uploading solutions. Uploading an incorrect solution is worth **-5** points. This penalty is additive, if

you upload more incorrect solutions, you will receive it multiple times. For some problems, we distinguish format errors (unparsable outputs) from incorrect outputs, and the former will not be penalised.

Please note that there is no point in uploading another solution for an already solved testcase because you cannot achieve more points with it. Therefore the system will not register additional uploads for solved testcases.

Additional information for scaled problems:

In this case there will be no penalty for uploading a solution later, so you are able to achieve the maximum amount of points by submitting in the very last minute - if you beat the other teams' solutions, that is.

Scores to scaled problems are recalculated occasionally (every few minutes). Your points may decrease in time (when another team submits a better solution than yours).

Please be aware that only your last submission is considered - not your best one.

Good luck and have fun!

About the Submission site

The location of the submission site is:

<http://ch24.org/sub>

You will be able to log in to the submission site with your registered team name and password. After login you can access three main views:

Team Status

You can see your team's status here, with all your submissions and the points received for them.

Submit

This is where you can post your solution files. You can upload multiple output files for multiple problems with a single submit. The naming of the output files must strictly match the following format: X99.out - where X is the problem's character code followed by a number (1 or 2 digits) identifying the test case.

Scores

Here you can see the current standings of the contest. This will not be available in the last hour.

Contact

You should subscribe to the public mailing list at <http://lists.ch24.org> to receive announcements and to be able to send feedback. The address of the list is `ch24@ch24.org`.

During the contest we will be available on IRC on the `irc.ch24.org` server (using the default port, 6667). For general discussion about the contest use the `#challenge24` channel, for problem specific discussion use `#p`, `#q`, `#r`, `#s` channels.

P. Pow

Your task is to evaluate a mysterious function on various inputs. The function is given by the following pseudocode:

```
function magic(X, Y):
  R := 1
  while Y > 0:
    R := R * X
    Y := Y - 1
  return R mod 2011
```

all the variables are non-negative integers, the operators have the usual semantics.

Input

Each input will contain several test cases, one test case per line.

Each test case consists of two positive integers: X,Y

The last line of the input will contain two zeros.

Output

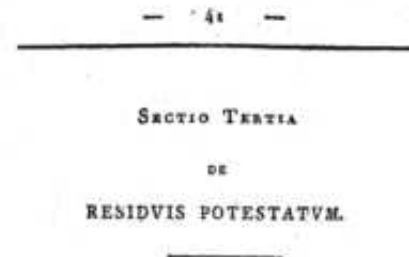
For each test case print one line with the return value of magic(X,Y)

Example input

```
1 24
10 2
2012 1
0 0
```

Example output

```
1
100
1
```



45. THEOREMA. In omni progressionis geometrica, $1, a, aa, a^3$ etc. propter primum 1 , alius ad hoc datur terminus, a^p , secundum modulum p ad a primum unitati congruus, cuius exponent $1 < p$.

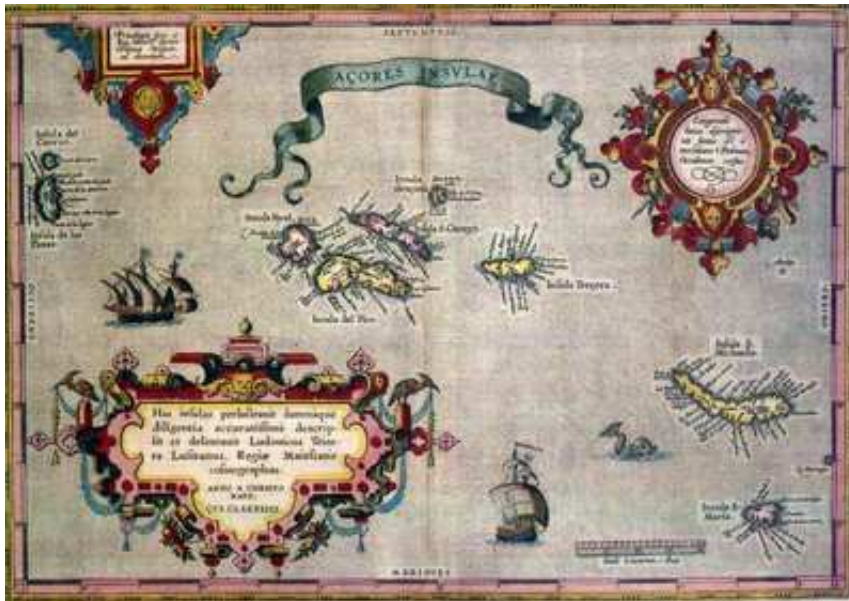
Demonstr. Quoniam modulus p ad a , adeoque ad quamvis ipsius a potestatem est primus, nullus progressionis terminus erit $\equiv 0 \pmod{p}$, sed quibus alicui ex his numeris $1, a, a^2, \dots, a^{p-1}$ congruus. Quorum multitudo quam sit $p - 1$, manifestum est, si plures quam $p - 1$ progressionis termini considerentur, omnes residuis minima diversa habere non posse. Quocirca inter terminos $1, a, aa, a^3, \dots, a^{p-1}$ bini ad minimum congrui inveniuntur. Sit itaque $a^m \equiv a^n$ et $m > n$, fietque dividendo per a^n , $a^{m-n} \equiv 1 \pmod{p}$ (art. 22) ubi $m - n < p$, et > 0 . Q. E. D.

Ex. In progressionem $1, 2, 4, 8$ etc. terminus primus qui secundum modulum 15 unitati congruus est 2^4 .

C 5

Q. Treasure island

Badarg, the great pirate got his hands on a map of a treasure island. The island has a rectangular shape and it is full of treasures. Unfortunately he is not the only one who knows about the island so he has to hurry to get to the island first and then leave before others get there.



The map is a $W \times H$ sized grayscale image where each pixel gives the value of a treasure item hidden at that location.

Due to the economic recession, Badarg has only a small boat and a small backpack so he can only carry 512 treasure items. He does not want to be on the island for more than a day and he can only walk about 4096 units distance a day. (1 pixel on the map is 1 unit)

So he will start from some X_0, Y_0 point which is on the border of the island and leave his boat there. Then he is going to visit a few locations on the island and pick up the treasures there. In the end he must go back to his boat.

Now he asks your help to prepare a treasure hunting plan for him, he even generously offers you the 1% of the looted treasures.

Input

The input is a $W \times H$ sized grayscale png image. The value of each pixel is between 0 and 255.

Output

The first line of the output must contain two integers, $X_0 Y_0$, the coordinates of the starting location on the border of the map.

The following lines must contain the $X Y$ coordinates of the visited locations in order, where $0 \leq X < W$ and $0 \leq Y < H$. At most 512 locations can be visited.

The distance is calculated by adding together the euclidean distance of the consecutive visited locations (including the distance from X_0, Y_0 to the first and last visited locations).

The maximum distance constraint will be checked with 0.1 unit tolerance.

Note that once an item is picked up somewhere, visiting the same location again Badarg won't find anything valuable there.

Score

$$\text{SCORE} = 100 * (1 - \text{sqrt}(1 - \text{SUM} / \text{MAXSUM}))$$

SUM is the sum value of the picked up treasures. MAXSUM is the best solution submitted so far. The score is recalculated at regular time intervals.

Example output

```
0 13
4 13
4 10
```

Starting from the left side of the map two locations are visited. The walked distance is 12.

R. Wav

We managed to record some communication fragments between alien spaceships. Our measurement device recorded 8000 samples a second. A sample is a real number between -1 and 1.

Your task is to decode the messages.

It is well known that aliens speak english between each other (see various movies).

Input

The first line contains N , the number of samples. Each of the following N lines contain a sample.

Output

English text.

Example input

First 3 lines of an input:

```
12345
0.1
-0.00789
```

Example output

```
hello world
```



S. Loopz

Badarg, the fearless pirate, was attacked by aliens on his way back from the treasure island. He can only escape by assembling his raft, stored in the captain's cabin in a big, gilded box. His raft is made from PVC pipe sections, which can be connected to form a single rectangular waterproof loop.

The raft is made of pipe segments, arranged in $x*y$ cells. There are 6 different kind of pipe segments (4 corners, a horizontal and a vertical).

A pipe section contains connected pipe segments that form a pipeline that has exactly two endpoints, and those endpoints are not blocked by other segments in the section. Badarg finds a set of such pipe sections in his box. His task is to arrange and place them (by moving and rotating the sections) in a way that:

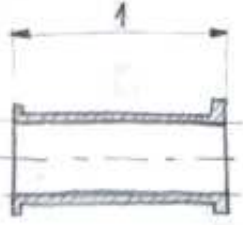
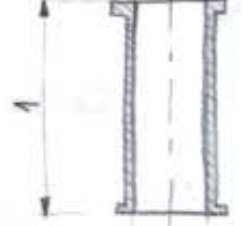
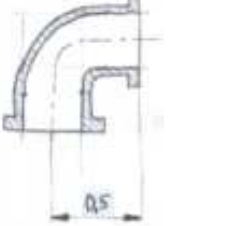

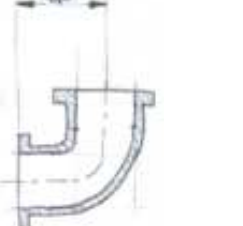
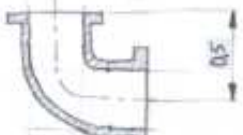
1. the final setup is a single loop of pipes.
2. there must be exactly one pipe segment in each cell.

Every input has a solution that uses all the pipe sections.

Input

The first line contains three integers: x , y , number of pipe sections. Then the pipe section descriptions follows. Each pipe section starts with a width,height line, followed by a 2d matrix of pipe segments.

A pipe segment is identified with an integer, 0 means empty tile, the meanings of the other identifiers can be seen in the table below:

| id | drawing | ascii | description |
|----|---|-------|---------------------|
| 1 |  | --- | horizontal pipe |
| 2 |  | | vertical pipe |
| 3 |  | — | top left corner |
| 4 |  | — | top right corner |
| 5 |  | — | bottom right corner |
| 6 |  | — | bottom left corner |

Output

The output should contain a line for each pipe section with three integers describing three operations:

- a rotation ($r = 0..3$); clock-wise, $r*90$ degrees
- a horizontal displacement of the left column of the section from the left edge of the map
- a vertical displacement of the top row of the section from the top of the map

(Badarg wears an eyepatch and cannot think in 3D - flipping the sections doesn't occur to him.)

Example input

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

Example output

```
1 0 0
0 1 1
1 3 1
```