



Preliminary Electronic Contest

12th BME International 24-hour Programming Contest

<http://ch24.org>



Preliminary Electronic Contest

Welcome to the testing round of the 12th BME International 24-hour Programming Contest!

This document is the problem set for the Preliminary Electronic Contest to be held on February 18th, 2012.

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 three 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 three 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 and Q 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 R 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 for the non-scaled problems 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 for those tasks.

Additional information for scaled problems:

In this case there will be no score 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. However, to avoid overloading our server, before submitting an updated solution for an input we will apply a delay. The delay is traced per team per task per input.

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` channels.

P. Independent

The princess of PonyLand would like to play with her friends (she has lots of friends), but she is a peace loving creature so she does not want to invite ponies who hate each other (those usually end up quarrelling).

Fortunately recent advances in online social networking in PonyLand produced a hugely successful anti-social website that collects data about which ponies hate each other. Of course the princess cannot simply access to this data due to proper privacy regulations in PonyLand, but she is in negotiations with the anti-social website owners.

While the negotiation is going on, an anonymized version of the anti-social graph of her friends is available.

Preparing for the party, the princess would like at least to know how many ponies she can invite at most.



photo by Freyda-s
source:
<http://www.flickr.com/photos/22329928@N08/4434576086>

Input

The input is the anonymized anti-social graph. First line contains N and M the number of vertices and edges, the following M lines contain two numbers between 0 and N-1, an edge of the graph.

It turns out that the anti-social graph in PonyLand has no cycles in it.

Output

The output is the number of ponies the princess can invite at most such that none of them are connected in the anti-social graph.

Example input

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

Example output

```
3
```

Q. Soundplot

You receive a set of encrypted audio messages from a secret admirer. Listening to the sounds yields nothing but a headache, but you have a knack that you would get something intelligible if you plotted each message as a drawing.

Input

The messages are all stereo .wav files. Each channel carries an incessant beeping sound that varies in pitch. You should draw a 2D plot, using the frequency of the left channel as the X coordinate, and the frequency of the right channel as the Y coordinate. (The frequencies map linearly to euclidean space.)



Output

For each input, you should send the message that you can (or can not) read on the resulting plot. Capitalization and punctuation do matter, as does the placement of whitespace (but the count and kind of whitespace doesn't).

Score

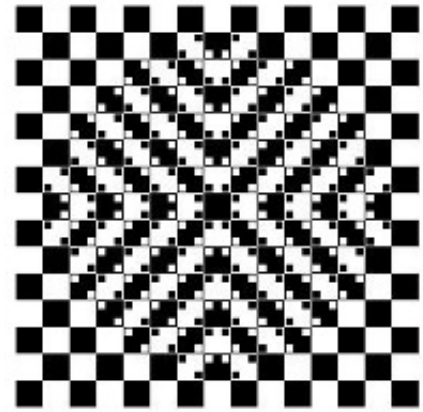
This task has exact scoring - you either get a submission right, or don't. Correct submissions are worth 100 points - incorrect submissions are penalized -5 points.

R. Squares

Your third IT startup having failed in a row, you have no other choice but to seek a job at Mandatory Suit Corp. On your first workday, your task is to optimize the output of some cryptic, unlabelled sales diagrams.

As it happens, you must use the in-house Mandatory Suit Graphics Library. This library lacks any sort of documentation, but you're a resourceful person, and reverse-engineered the usable API. This wasn't very difficult, though - apparently there is only one call, and that draws 1px thick, axis-aligned black unfilled rectangles.

Your task is to analyze the input images, and output a list of calls that draw the same image, using the least possible amount of drawing operations.



source:
<http://www.optical-illusionist.com/illusions/bulging-squares>

Input

Each input is an arbitrarily sized png image that contains black (RGB 0,0,0) and white (255,255,255) pixels.

Output

The output is a text file that contains:

- the coordinates of two pixels on each line
- each line is 4 integers separated by whitespace, then an newline
- in the following order: **X1 Y1 X2 Y2**
- the range of X coordinates is **0 .. width-1, from left to right**
- the range of Y coordinates is **0 .. height-1, from top to bottom**

Beginning with a white image, each output line draws a black unfilled rectangle between the specified pixels. After drawing all rectangles, the image must be identical to the original input.

We will not accept outputs that contain more lines than the number of pixels in the source image or more bytes than 10000000.

Score

This task uses scaled scoring. For each input, the team with the shortest correct solution gets 100 points (as well as all other teams with solutions that are the same length). Longer solutions get progressively less points.