



6th BME-KTH International
24-hour Programming Contest



Qualifying Round
Problem Set

February 26, 2006

<http://www.challenge24.org>



FORNAX

The main sponsor of the event is Fornax Co.

GRAPHISOFT®

Virtual Building Solutions

International Golden Grade Sponsor

evosoft  SIEMENS

National Golden Grade Sponsors



Silver Grade Sponsors



John von Neumann Computer Society, IEEE, HTE
Professional Sponsors

6th International 24-hour Programming Contest

Qualification Round

General remarks

- There are six problems: A, B, C, D, E, and F.
- For each problem there are 10 input files (each input file is called a test set), numbered from 0 to 9.
- To solve a test set, you can use any platform, operating system, programming language, development tools, libraries, etc. but you *may not use any external human resources*.
- For some problems, input and output will be in PNG format (24-bit colors). For images, pixel colors are referred to as triples (R, G, B) , giving the red, green and blue components of the color, respectively. Each component is an integer between 0 and 255 (inclusive).
- For problems with text input, the $i + 1$:th test set for problem X is named X- i .in (e.g. C-3.in for the fourth test set to problem C). For problems with image input, the $i + 1$:th test set for problem X is named X- i -in.png (e.g. F-9-in.png for the tenth test set to problem F).
- We have provided some sample input/output files for the problems, located in the `samples` directory. We suggest that you check your program with the sample files first, and submit your solution only if your output matches the sample outputs.
- You can submit your solutions on <http://ch24.csc.kth.se>, where you have to log in using your team name and password. There you can also find the ranklist.
- The 30 teams scoring the highest points will gain the right to participate in the final at Budapest, Hungary April 21 – April 23, 2006.

Scoring

The scoring mechanism of this contest is slightly complicated, and works as follows:

- A submission for a test set is assigned a base score between 0 and 100. This base score is problem-specific, and will be further described in each problem. Furthermore, this base score may be relative to the base score of other teams' submissions on this test set, and thus the base score of your submission may decrease later.
- Based on the base score of your submission, you get a certain score. If your S :th submission to a test set is submitted T minutes into the contest, getting a base score of X , then the score of this submission will be $X \cdot 0.98^s \cdot 0.98^{t/20}$, rounded to the nearest integer. Here,
 - $s = \max(0, S - 3)$. In other words, after your third submission to a test set, the test loses 2% of its value for every additional submission.
 - $t = \max(0, T - 60)$. In other words, after the first hour, each test case loses 2% of its value for every 20 minutes.

Your score for a test set will be the maximum score of any of your submissions to this test set.

For example:

- If your 5:th submission for a test set is submitted 2 hours and 40 minutes into the contest and getting a base score of 80, your score for this submission will be $80 \cdot 0.98^2 \cdot 0.98^5 \approx 69$ points.
- If your 8:th submission for a test set is submitted 3 hours and 55 minutes into the contest and getting a base score of 93, your score for this submission will be $93 \cdot 0.98^5 \cdot 0.98^{8.75} \approx 70$ points.

Problem A: An Optimal Keyboard

You have probably heard about the Dvorak keyboard layout, which is more efficient than the traditional QWERTY layout. The Dvorak layout places commonly used (for English text, that is) letters on the home row and in easily reached positions. However, for other types of text, it may be suboptimal. For instance, the commonly used Unix command 'ls' is entirely typed by the right little finger.



There is a new type of keyboard where all the keys have OLED displays. Using this technology, we could create a keyboard where the layout adapts to the text you are typing. Your assignment is to work on a small part of this system, the part that, given the text that the user will be typing, calculates an optimal character placement. You do not need to worry about how the program will know the text the user will be typing, we're working on an advanced brain-computer interface to solve that minor issue.

For each key, we will assign a numeric value in the range [0-9] which measures how difficult that key is to reach. We will define the total difficulty of typing the text as the sum of the difficulty for each key pressed to produce the text.

For example, to type the text 'abcDAA' with the character 'a' in a position with difficulty 1, 'b' and 'c' with difficulty 8 and 'd' with difficulty 9 and shift in a position with difficulty 1, the difficulty of typing the text would be 31 (to produce an uppercase 'A', the difficulty is 2, 1 for the shift and 1 for the 'a' character). Keys can not be held down, so the cost for producing the 'AA' part is 4 (pressing shift twice and 'a' twice).

There are three special characters, space (s), enter (e), and shift (h). In the output keyboard layout, (English) letters are written in uppercase. The shift key only affects (English) letters, so to produce a left parenthesis ('('), that character must be mapped to a key on the keyboard, it can't be produced by shift-9 (as on an American keyboard layout). Each character may *only* appear once on the keyboard layout. It is acceptable for unused characters to appear on the keyboard layout, and it is also acceptable to put 'x' on keys to indicate that that key is not mapped to any character.

Legal characters to place on the keyboard are the (ascii) characters in the (inclusive) range 33-126 ('!' through '~'), except for lowercase letters (except for 's', 'e', 'h', and 'x', as noted above).

Input

Each test set will contain a keyboard template and the text to be typed. The first line of the keyboard template will consist of an integer specifying the number of lines of the keyboard template. The next lines will consist solely of characters [0-9] and 'x', where an integer denotes that a character could be placed in that position, with that difficulty, and 'x' denotes that a character can't be placed there.

After the keyboard layout, the text to be typed will follow. Please note that the last character of the input will always be a newline character, which should be processed.

Output

The first line of output should be the total difficulty of typing the text using your keyboard layout. On the following lines, a keyboard specification should be printed, in the same format as in the input (i.e. first a line specifying the number of lines to follow), but with the digits replaced by the characters you mapped there, or 'x' if you did not use that key. Letters are given in uppercase. The (lowercase) letters 's', 'h', and 'e' are used for the space, shift and enter keys.

Scoring

Your base score for a test case will be calculated as follows. Let Val be the total difficulty achieved by your solution, and Opt be the minimum total difficulty of the best solution of any team, including the judges' solution. Then the base score of your solution is given by

$$100^{2-Val/Opt}.$$

In other words, if your solution matches the best solution, the base score will be 100, and if your solution is twice as difficult as the best solution, the base score will be 1.

Problem B: Building Pipe Systems

In the country of Elbonia, the citizens are getting tired of having no sanitation system. Together, they have saved some money to build a new pipe system that will connect their houses to the communal sanitation system. But since this is a small country, where most of the people have small incomes, they really want to build the cheapest system possible. The largest cost in this whole endeavour will be the pipes that must be put in the ground, connecting all the houses. Therefore, they need your help in minimizing the total length of these pipes.

A pipe system will connect all the houses together. The houses may either be connected house to house, or they may be connected via arbitrarily positioned junction points. Since Elbonia consists entirely of mud, the cost of laying down the pipes is uniform over all the land. In other words, the only thing that matters is that the total length of pipes is as small as possible.

Input

Each test case contains the layout of one town. The consists of a number N , denoting the number of houses. Following this number come N lines, each containing a pair $x_i y_i$ denoting the position of one house. The values x_i and y_i are both floating point numbers of single precision length.

Output

For each test case, you will need to produce an output that represents a fully-connected pipe system. The smaller the total length of this pipe system, the higher the value of your solution will be. The representation of this pipe system should consist of the following elements.

On the first line, a number M , where $N \leq M < 10 \cdot N$, denoting the number of points on the pipe system. Following this come M lines, each containing a pair $x_i y_i$ denoting the position of one house or one junction. The first N lines should contain the house locations from the input in the same order, while the next $M - N$ lines contain your choosen junction positions.

Following this, on a new line, the number K of pipes that your pipe system has. On the following K lines, you should print pairs $a_j b_j$ of numbers, where $0 \leq a_j, b_j < M$ and $a_j \neq b_j$, specifying the points connected by the pipe. Furthermore, each pair $a_j b_j$ may only occur once in the pipe system. The indexing of the points is zero-based.

Scoring

Your base score for a test case will be calculated as follows. Let Val be the length of your solution, and Opt be the length of the best solution of any team, including the judges' solution. Then the base score of your solution is given by

$$100^{2-Val/Opt}.$$

In other words, if your solution matches the best solution, the base score will be 100, and if your solution is twice as long as the best solution, the base score will be 1.

Almost blank page

Problem C: The Civilised Cells' Civil War

Somewhere, in a human body not far away from here, a peculiar war is taking place. The war stands between two different kinds of cells, A-cells and Z-cells, both striving to subdue the other in order to gain power, money, and better health benefits.

We model a battle between A-cells and Z-cells as being played on a two-dimensional grid, in which each grid square can be occupied by an A-cell, occupied by a Z-cell, or empty. Being quite the gentlecells, the cells take turns attacking each other. The A-cells always start the battle, since A is way earlier than Z in the alphabet. Each turn, one cell of the kind whose turn it is must attack a neighboring (either above, below, to the left, or to the right) enemy cell, thereby consuming it. The cell then moves to the consumed cell's position, leaving its old position empty. If no attack is possible (because there are no cells with an enemy cell next to it), the side whose turn it is withdraws from the battlefield in shame, and the other side wins.

Figure (a) below shows a possible battlefield, and figure (b) shows the result of the A-cells attacking the Z-cell in row 3, column 3. Note that this is not a very clever move, since the Z-cells can then attack using their cell in row 3, column 2, thereby winning the battle. Had the A-cells attacked the Z-cell in row 1, column 3 instead, they would have won.

A		Z	
		A	
	Z	Z	

(a) A possible battlefield

A		Z	
	Z	A	

(b) The result of A moving downwards from row 2 col 3

The increasingly desperate Z-cells have realised that they need some real brainpower in order to be able to win this war. Thus, they have turned to the brain and asked it to help them fight the A-cells and win the battles. In turn, the A-cells, realizing that they too will need an ally when faced with this new opponent, have sided with the mouth, which on many people operates independently of the brain. Guided by these two masterminds, both sides will fight their battles optimally. In other words, if it is possible for any side to make moves in such a way that they will win the battle no matter what the other side does, they will do that.

Now, your task is to help us determine what the outcome of some battles will be, given the layout of the battlefield at the start of the battle.

Input

Each test set consists of several battlefields to be analyzed. The first line of each input file consists of an integer N , indicating the number of battlefields to be analyzed. Then come the N battlefields, each described as follows:

The first line of each battlefield consists of two integers W , H , the width and the height of the field, respectively. Then follow H lines, each consisting of W characters, describing the battlefield. Each character is one of 'A', 'Z' or '.', indicating whether that position is occupied by an A-cell, occupied by a Z-cell, or empty, respectively.

Output

For each battlefield, output one line containing the character 'A' or the character 'Z', indicating who will win this battle.

Scoring

Your base score for a test set will be $100 \cdot 10^{-8k/N}$, where k is the number of battlefields for which your answer was *incorrect*, and N is the total number of battlefields in the test set. For example, solving all battlefields correctly gives a base score of 100, solving 75% of the battlefields correctly gives a base score of 1, and solving half the battlefields correctly gives a base score of 0.01 (and thus will not give any score).

Problem D: Dangerous Currents

You are planning a canoe field trip, and you have planned everything, from how to make your digital equipment water resistant, to the auto pilot system. You have obtained maps of the most difficult parts of the rivers. The maps show how deep the water is, and you estimate that the maximum amount of water that may flow through a point is approximately proportional to the water depth around that point. Now you want to calculate how much water can flow in a certain segment of the river, assuming that there is no limitation in how much water may come from upstream, and how much water may continue downstream. You have arranged the maps so that the river flows from left to right in all pictures.

Input

Each test set is a PNG image, depicting the river. The color of a pixel is either of the form $(127 + h, 127 + h, 127 + h)$, indicating that this point is land of height h , or of the form $(127 - d, 127 - d, 255 - d)$, indicating that this point is water of depth d . For land pixels, $0 \leq h \leq 127$, and for water pixels, $1 \leq d \leq 127$. The maximum amount of water, measured in cm^3/s , that can flow between two vertically or horizontally adjacent pixels is the minimum of their depths.

Output

The output should be an image with the same dimensions as the input image. In this image you should mark a possible way for water to flow through this part of the river. The higher the total amount of flow in your answer, the higher base score your answer will get.

In the output, the color of land pixels should remain unchanged, and water points with depth d should have the color $(128 + y, 128 + z, 255 - d)$, where y is the amount of water that flows downwards from this point, and z is the amount of water that flows to the right from this point (measured in cm^3/s). Note that if water flows *into* this point from below, y will be negative, and similarly for z .

Scoring

Your base score for a test case will be calculated as follows. Let Val be the total water flow in your solution, and let Opt be total water flow in the best solution of any team, including the judges' solution. Then the base score of your solution is given by

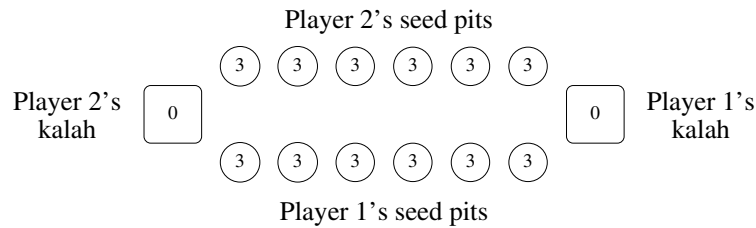
$$100^{2 - \text{Opt}/\text{Val}}.$$

In other words, if your solution matches the best solution, the base score will be 100, and if your solution only has half as much flow as the best solution, the base score will be 1.

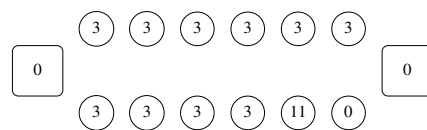
Almost blank page

Problem E: Evaluating Kalaha Boards

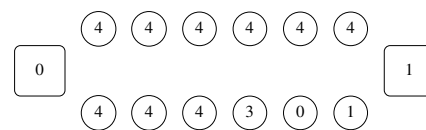
Kalaha, sometimes called Kalah, or Mankala¹ is a game between two players. It is played on a board where each player has a number of seed pits (usually six), and a seed store, called a kalah. The pits and kalahs are organised in a circular fashion as displayed in the figure below.



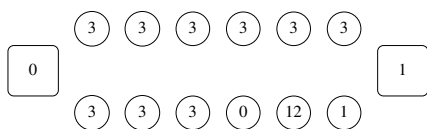
Initially, each pit is filled with a certain number of seeds (usually three). A player makes a move by choosing a seed pit and then redistributing the seeds from that pit, to the following pits and her/his own kalah, in counterclockwise order (see Figure (b) below). If the last seed was placed in the player's kalah, the player continues by choosing a new seed pit (so after making the move depicted in Figure (c) below, the player will move again). If the last seed was placed in one of the player's own empty seed pits, that seed, and all seeds in the opposite seed pit, are moved to the player's kalah (see Figure (d) below).



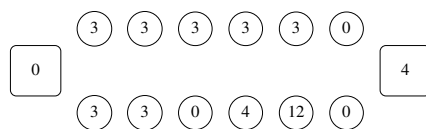
(a) A Kalaha board



(b) The result of choosing pit 5 in (a)



(c) The result of choosing pit 4 in (a)



(d) The result of choosing pit 3 in (a)

The game ends when one of the players has no more seeds in her/his seed pits. Any remaining seeds in the other player's seed pits are then moved to her/his kalah. The winner is the player with the most seeds in their kalah.

Your task is to, given a Kalaha board, find out how many seeds the current player can get during her/his turn (note that this includes seeds gained in any additional moves that can be made in case a move ends by placing a seed in the player's kalah).

Input

Each test set consists of a Kalaha board to be analyzed. The first line of a test set contains an integer n , the number of seed pits of each player. Then follow $2n$ integers. The first integer gives the seed count of the current player's first seed pit, then comes the seed count of the remaining seed pits, in counter-clockwise order (i.e. the n :th integer gives the number of seeds in the current player's last seed pit, and the $2n$:th integer gives the number of seeds in the opposite player's last seed pit).

¹Though Mankala is actually the name for the family of games to which Kalah belongs.

Output

The first line of output should consist of a line containing two integers M , S , the number of moves you make this turn, and the number of seeds you manage to place in your own kalah. Then there should be M lines, the i :th containing an integer j between 1 and n , indicating that you choose seed pit j for the i :th move. Every move but the last must end in with a seed being placed in your kalah, and the last move may not end with a seed being placed in your kalah, except if the game has ended.

Scoring

Your base score for a test case will be calculated as follows. Let Val be the number of seeds you place in your own kalah, and let Opt be the maximum number of seeds for this board achieved by any solution, including the judges' solution. Then the base score of your solution is given by

$$100 \cdot 10^{1-Opt/Val}.$$

In other words, if your solution matches the best solution, the base score will be 100, and if your solution only captures half as many seeds as the best solution, the base score will be 10.

Problem F: Floodland

This problem takes place a long time ago, slightly before the invention of water. At this time, the world was entirely dry, but this was about to change: rumor had it that the powers that be were planning to flood the world with something called “water”. This water thingy was reported to be pleasant, but not really something you would want to live in. Noah, an enterprising young real estate agent realised that big business could be made by finding out where there would be water, and more importantly, where there would *not* be water, after the flood. Given a map of the world, your job is to help Noah with this task.



You will be given a height map, in the form of an image, giving the height of the land at different points on the earth. Water can flow between horizontal or vertically adjacent points on this map, as long as does not flow upwards. Depending on the surface of the land, water will collect in lakes and oceans, closed regions surrounded by higher land. Remember that at this time, the world was still flat (ignoring the variations in the landscape), not spherical. Thus, any water reaching the edges of the world will pour off the world, never to be seen again.

Input

Each test set is a PNG image. The color of any pixel is of the form (h, h, h) , indicating that the height of the world at this point is h .

Output

The output for a test case should be the same image, but for points where there a lake will be formed, the color should be changed to $(h, h, 255 - d)$, where h is the height of the land, and d is the depth of the lake at this point.

Scoring

Your base score for a test set will be $100 \cdot 10^{-8k/N}$, where k is the number of pixels in your output image that are *incorrectly* colored, and N is the total number of pixels in the image. For example, getting the entire image correct gives a base score of 100, getting 75% of the image correct gives a base score of 1, and getting half the image correct gives a base score of 0.01 (and thus will not give any score).