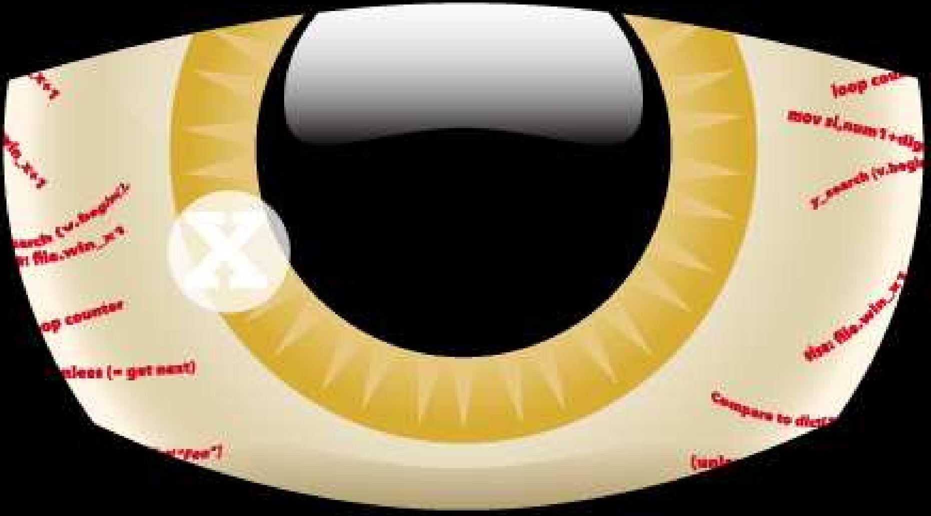
[redacted]



[redacted]

[redacted]

Test Competition

10th Jubilee BME International 24-hour Programming Contest!

www.challenge24thelegend.com

Test Competition

Welcome to the test round of the 10th BME International 24-hour Programming Contest!

This document is the problem set for the Test Competition to be held on February 20th.

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

The three problems we selected for the Test Competition may not be the kind of problem you'd consider especially difficult, but they may still provide some clues about the Electronic Contest - the tools you'd find useful, the usage of our result submission system, and so on.

Rules

The Test Competition contains 3 problems. You have all the time in the world to solve them, but we take submissions from 08:00 to 20:00 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://x.challenge24thelegend.com/ch24ECServer. Your solutions will be evaluated on-line.

There are two major problem types:

- Problems that have an exact solution will be evaluated almost immediately, and a final score will be given for uploaded outputs. We will refer to these problems as "non-scaled problems". In the Test Competition, **P** and **Q** are non-scaled problems.
- Problems that do not have a known "best" solution will be evaluated periodically and the score is scaled according to best ever uploaded output. In this case you will compete directly against the other teams. We will assign a numeric value to each uploaded solution and in every couple of minutes calculate the actual scores from them. We will refer to these problems as "scaled problems". In the Test Competition, **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. Achievable maximal points can differ per test case, these will be indicated at the problem descriptions.

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 **-10** points. This penalty is additive, if you upload more incorrect solutions, you will receive it multiple times.

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 maximal points by submitting in the very last minute, if you beat the other teams' solutions.

When you upload a solution for a scaled problem, your summed points will temporarily decrease, until the next scaling occurs, when you will receive your points. Also be aware that your points might decrease in time when another team submits a better solution than yours.

Please be aware that your last submission will count which is not necessarily your best.

Tracing

For certain problems we provide you a way to get more detailed feedback (a trace or log of the evaluation) for your solution. Unless otherwise stated in the problem description, you will receive a **-10** point penalty for each trace you ask. Because you can upload multiple solutions together, if you ask for tracing for one of your submits, all solution files in the zip file you upload will be traced that belong to a traceable problem. The tracing penalty is additive as well, so be careful about your uploads when you ask for detailed feedback.

Upload frequency:

In some cases we define a minimal time gap between two submissions. This means that you are not able to upload two solutions for the same problem within this minimal time gap. This is applied for different test cases of the same problem as well.

Good luck and see you in the finals!

About the Submission site

The location of the submission site is:

http://x.challenge24thelegend.com/ch24ECServer

You will be able to log in to the submission site with your registered email address and password. After login you will see your login email and team name in the top right corner. You can submit solutions by logging in with any of your team members' account. You can also use multiple accounts to submit the solutions. The site has five pages:

Announcements

This page will show the announcements made during the contest. These can be general information, or something connected to a problem (e.g. if an ambiguity is found in the description, we will correct it by posting an Announcement).

Team status

You can see your team's status here, with all your submissions and the points received for them. If you click on the points, you will be able to download the file you have submitted. Also this is the page where you can download the detailed logs (traces) of those submissions that you have asked trace for.

Submit solution

This is where you can post your solution files. You have to upload zip files, that contain one or more output files. 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. You may ask for a trace by checking the checkbox here. You can download the detailed log later. Asking for trace is applied to ALL solutions submitted in the same zip file.

Standings

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

Problem information

Here you can see a summary of the various parameters of each problem

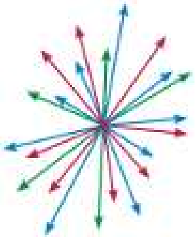
- Problem name and code
- is it a "scaled problem"
- can you ask for a trace
- minimum time between two uploads for this problem

Also you can see the maximum points that you can get for each testcase if you upload a correct solution at that time.

Contact

During the contest we will be available on IRC on the `irc.challenge24thelegend.com` server. For general discussion about the contest use the `#challenge24` channel, for problem specific discussion use `#p`, `#q` and `#r` channels.

P. Normalize



In this task, you have to normalize a sequence of real numbers.

Input

The first line of the input contains the number of elements in the sequence (N). The next N lines each contain an element of the sequence.

Output

The output should have N lines containing the same sequence of numbers as the input, but scaled with a constant, such that the sum of the squares of the numbers give 1. We will check the results with 0.001 tolerance. (The ratio between the numbers in the input and output and the sum of the squares of the numbers are checked).

Example

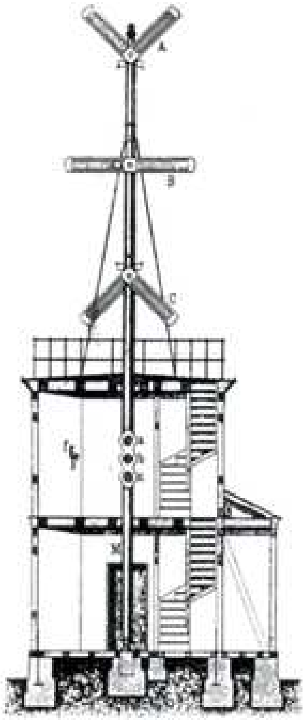
in:

```
3
-0.834065
-0.0092335
0.10021
```

out:

```
-0.9928
-0.0109908
0.119281
```

Q. Absurdistanian ISP



Finally, after the previous government has fallen, it is now possible to get Internet access in Absurdistan. The economical and political turmoil in the country is so severe that only one Internet Service Provider dared to set foot in the frontier market - a local company, composed of the brightest and most enterprising engineers the country had to offer.

Regrettably, education in international telecommunication isn't yet up to speed in Absurdistan. Lacking knowledge of today's widespread technologies, the engineers came up with their own communication framework.

In this system, data exchange is applied through the existing analog phone lines - using the actual phones themselves, with the PC's sound card connected to the phone via a proprietary microphone adapter sold by the ISP (Absurdistan's marketing and management schools lead the way in modernization). Data encoding is simple: everything is transmitted as a series of short and long beeps.

Users are dissatisfied with the existing sound decoder software, because it's slow, insensitive, and tends to insert random pieces of political propaganda in its output. Your task is to rewrite the decoder and test it on the provided set of sound samples. Each wav file encodes a single sentence written in uppercase English. Output should be the decoded text file. (When validating outputs, we will ignore empty lines and white space differences.)

Dot is a short beep and dash is a long beep. There is always a short gap between two beeps and a longer gap between two letters. Space is represented by an even longer gap. Since the Quality Assurance Department of the ISP is only barely existent, the timing and signal frequencies vary by city, weather, time of day and moon phase.

Input

Input consits of digitized sound files in wav format. The audio is sampled at 18000 Hz frequency using 8 bit samples and it is given as mono, uncompressed wav. (Note that Wav headers can be easily skipped by ignoring the first 44 bytes and reading the raw data directly; each byte of raw data is an unsigned 8-bit integer.)

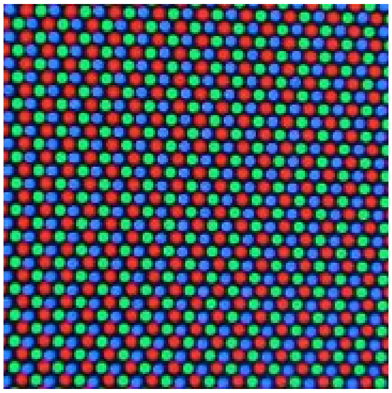
Output

Plain text written in English.

Code Table

Letter	**Encoding**	**Letter**	**Encoding**
A	. -	S	. . .
B	- . . .	T	-
C	- . - .	U	. . -
D	- . .	V	. . . -
E	.	W	. - -
F	. . - .	X	- . . -
G	- - .	Y	- . - -
H	. . . .	Z	- - . .
I	. .	,	- - . . - -
J	. - - -	0	- - - - -
K	- . -	1	. - - - -
L	. - . .	2	. . - - -
M	- -	3	. . . - -
N	- .	4	. . . . -
O	- - -	5	. . . . .
P	. - - .	6	- . . . .
Q	- - . -	7	- - . . .
R	. - .	8	- - - . .
		9	- - - - .

R. Columns



Archeologists have recently found unquestionable proof that some ancient cultures used a complex method of encoding information in pictures. The ancient artists hid their arcane message in a few pixel columns of their pictures, selected according to a secret formula known only to a few. Fortunately, our archeologists managed to find the formula on ancient stone tablets. Even with the formula, finding the proper columns is a non-trivial task - only when the proper columns were found by luck on the flaking paint over an ancient egyptian grave, it became clear that this code is part of something big. A great mistery is about to be solved - but first, the columns must be found.

In this task, your input is a JPG image file. You have to select five columns of this image and send back the index of these columns. We will evaluate our magic expression (see below) on the selected five columns, resulting in a number. The goal of the task is to minimize this number. For each input file, the team with the smallest result will get the most points.

The Expression

For each row in the image, we will take the colors of the pixels in the given 5 columns. The red, green, and blue components are used, mapped from 0 to 255. Then we will calculate the value of the following expression:

```
((red1 - green2) * red3 + green4) * blue5
```

(Where red1 is the red component from the pixel in the first column, green4 is the green component from the pixel in the fourth column, and so on.)

The result of this expression for each row will produce a total sum; we will then divide this total sum by 8 and take the absolute value to get the final result.

Input

The JPG image file.

Output

Five integers of the selected five columns. The leftmost column has the index 0. The output should be written as a single line of text, five numbers separated by whitespace.

Example

Say we have the following 5x4 pixel input image (written as R, G, B):

```
  col0      col1     col2      col3      col4
 0, 0, 0   8, 2,4   0, 8, 5   3, 5, 4   0, 0, 0   row0
 0, 0, 0   4, 5,7   0, 0, 0   7, 1, 1   0, 0, 0   row1
 1, 1, 3   7, 2,6   0, 0, 0   4, 4, 6   0, 0, 0   row2
 0, 0, 2   7,10,3   3,11, 0   3, 4, 6   0, 0, 0   row3
```

With the following submission:

```
0 1 2 3 3
```

Then the score will be evaluated as follows:

```
  ((0 -  2) * 0 + 5) * 4
+ ((0 -  4) * 0 + 1) * 1
+ ((1 -  2) * 0 + 4) * 6
+ ((0 - 10) * 3 + 4) * 6
```

Total sum will be `-111`; absolute value of `-111/8` with integer division will give the final score `13`.