CODEFORCES β
Sponsored by Telegram

Huawei Honorcup Marathon 2

A3. Shuffled Images

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this problem, you have to restore the image, if it was previously divided into square fragments of $p \times p$ pixels, which were shuffled randomly. You are given a set of images, each has a size of 512×512 pixels. Divide each image into $m \times m$ square fragments of $p \times p$ pixels ($m = 512/p$) and rearrange them to restore the original image. It is guaranteed that all original images are regular photos (perhaps with some minor vertical/horizontal scales).

For each image, you can get from 0 to 100 points. The score for the image is equal to the ratio of correctly defined "edges" between the images. Here, by an "edge" we call a horizontal or vertical segment of length p pixels, which separates two adjacent fragments along the side. An "edge" is defined correctly if output contains the edge the answer (the order of the fragments "bottom-top" or "left-right" matters). The total number of such "edges" is $2 \cdot m \cdot (m-1) = k$. Formally, the score for the image will be $100 \cdot a/k$, where a is the number of correct "edges". Points for the images in a test are summarized. If you do not provide an answer to the image, then the score for it is 0.

Problem A contains tests with parameters $p = 64$, $m = 8$, $k = 112$. Problem B contains tests with parameters $p = 32$, $m = 16$, $k = 480$. Problem C contains tests with parameters $p = 16$, $m = 32$, $k = 1984$. Each problem is divided into three subproblems.

Input

The archive with data is available at the link https://icpc.sgu.ru/files/shuffled-images-data.zip or https://yadi.sk/d/BmnFhkaD1Vy4vA. The data is divided into three folders: `data_train`, `data_test1_blank`, and `data_test2_blank`. Each of the folders contains three subfolders: 64, 32 and 16. Each of the subfolders contains a set of png-files with fragment sizes 64, 32 and 16 respectively. You need to process all images (or any subset of them) and send a text file containing information about the permutations of fragments for each of the files.

Files in the `data_train` folder are intended for training. This folder contains additional files `data_train_64_answers.txt`, `data_train_32_answers.txt` and `data_train_16_answers.txt` containing the correct answers for all images in corresponding subfolder. Additionally, this folder contains subfolders `64-sources`, `32-sources` and `16-sources`, which contains source images for the corresponding set.

Images in the `data_test1_blank` folder are intended for preliminary testing, images in the `data_test2_blank` folder are intended for final (system) testing.

Output

A submission is a text file with the following contents:

```
file_name_1
permutation_1
file_name_2
permutation_2
...
file_name_n
permutation_n
```

where `file_name_i` is the name of the image file for which the answer is provided, and `permutation_i` is the permutation of integers from 0 to m^2-1. Enumerate the fragments in the image from left to right, top to bottom. The i-th number in the permutation should be equal to the fragment index of the given (i.e., shuffled) image, which is located in the i-th fragment of the resulting image.

All filenames in the text file sent out must be different and correspond to the image filenames in the corresponding folder. It is allowed not to provide answers for one or more images.

Problems A1, B1 and C1 contains tests from the `data_train` folder with fragment sizes 64, 32 and 16 respectively. Answers to the training data are open to you, but you can send your answers if you want. These tasks are only needed to test your solutions. These problems will be hidden before official system testing.

Problems A2, B2 and C2 contains tests from the `data_test1_blank` folder with fragment sizes 64, 32 and 16 respectively. Your solutions will be checked and evaluated immediately after each submission. Use these problems to check and estimate the quality of your approach. These problems will be hidden before the official system testing.

Problems A3, B3 and C3 contains tests from the `data_test2_blank` folder with fragment sizes 64, 32 and 16 respectively, by which the winners will be determined (official system tests). Your solutions will not be judged during the contest, but the format will be checked. A properly formatted solution will receive 0 points during the contest. After the contest, for each participant we will use the *last* correctly formatted (0 points) submission on problems A3, B3 and C3 to evaluate the quality of the solution. **These values will**

be used to determine the winners.

B3. Shuffled Images

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this problem, you have to restore the image, if it was previously divided into square fragments of $p \times p$ pixels, which were shuffled randomly. You are given a set of images, each has a size of 512×512 pixels. Divide each image into $m \times m$ square fragments of $p \times p$ pixels ($m = 512/p$) and rearrange them to restore the original image. It is guaranteed that all original images are regular photos (perhaps with some minor vertical/horizontal scales).

For each image, you can get from 0 to 100 points. The score for the image is equal to the ratio of correctly defined "edges" between the images. Here, by an "edge" we call a horizontal or vertical segment of length p pixels, which separates two adjacent fragments along the side. An "edge" is defined correctly if output contains the edge the answer (the order of the fragments "bottom-top" or "left-right" matters). The total number of such "edges" is $2 \cdot m \cdot (m-1) = k$. Formally, the score for the image will be $100 \cdot a/k$, where a is the number of correct "edges". Points for the images in a test are summarized. If you do not provide an answer to the image, then the score for it is 0.

Problem A contains tests with parameters $p = 64$, $m = 8$, $k = 112$. Problem B contains tests with parameters $p = 32$, $m = 16$, $k = 480$. Problem C contains tests with parameters $p = 16$, $m = 32$, $k = 1984$. Each problem is divided into three subproblems.

Input

The archive with data is available at the link https://icpc.sgu.ru/files/shuffled-images-data.zip or https://yadi.sk/d/BmnFhkaD1Vy4vA. The data is divided into three folders: `data_train`, `data_test1_blank`, and `data_test2_blank`. Each of the folders contains three subfolders: 64, 32 and 16. Each of the subfolders contains a set of png-files with fragment sizes 64, 32 and 16 respectively. You need to process all images (or any subset of them) and send a text file containing information about the permutations of fragments for each of the files.

Files in the `data_train` folder are intended for training. This folder contains additional files `data_train_64_answers.txt`, `data_train_32_answers.txt` and `data_train_16_answers.txt` containing the correct answers for all images in corresponding subfolder. Additionally, this folder contains subfolders `64-sources`, `32-sources` and `16-sources`, which contains source images for the corresponding set.

Images in the `data_test1_blank` folder are intended for preliminary testing, images in the `data_test2_blank` folder are intended for final (system) testing.

Output

A submission is a text file with the following contents:

```
file_name_1
permutation_1
file_name_2
permutation_2
...
file_name_n
permutation_n
```

where `file_name_i` is the name of the image file for which the answer is provided, and `permutation_i` is the permutation of integers from 0 to $m^2 - 1$. Enumerate the fragments in the image from left to right, top to bottom. The i-th number in the permutation should be equal to the fragment index of the given (i.e., shuffled) image, which is located in the i-th fragment of the resulting image.

All filenames in the text file sent out must be different and correspond to the image filenames in the corresponding folder. It is allowed not to provide answers for one or more images.

Problems A1, B1 and C1 contains tests from the `data_train` folder with fragment sizes 64, 32 and 16 respectively. Answers to the training data are open to you, but you can send your answers if you want. These tasks are only needed to test your solutions. These problems will be hidden before official system testing.

Problems A2, B2 and C2 contains tests from the `data_test1_blank` folder with fragment sizes 64, 32 and 16 respectively. Your solutions will be checked and evaluated immediately after each submission. Use these problems to check and estimate the quality of your approach. These problems will be hidden before the official system testing.

Problems A3, B3 and C3 contains tests from the `data_test2_blank` folder with fragment sizes 64, 32 and 16 respectively, by which the winners will be determined (official system tests). Your solutions will not be judged during the contest, but the format will be checked. A properly formatted solution will receive 0 points during the contest. After the contest, for each participant we will use the *last* correctly formatted (0 points) submission on problems A3, B3 and C3 to evaluate the quality of the solution. **These values will be used to determine the winners**.

C3. Shuffled Images

time limit per test: 1 second

memory limit per test: 256 megabytes
input: standard input
output: standard output

In this problem, you have to restore the image, if it was previously divided into square fragments of $p \times p$ pixels, which were shuffled randomly. You are given a set of images, each has a size of 512×512 pixels. Divide each image into $m \times m$ square fragments of $p \times p$ pixels ($m = 512/p$) and rearrange them to restore the original image. It is guaranteed that all original images are regular photos (perhaps with some minor vertical/horizontal scales).

For each image, you can get from 0 to 100 points. The score for the image is equal to the ratio of correctly defined "edges" between the images. Here, by an "edge" we call a horizontal or vertical segment of length p pixels, which separates two adjacent fragments along the side. An "edge" is defined correctly if output contains the edge the answer (the order of the fragments "bottom-top" or "left-right" matters). The total number of such "edges" is $2 \cdot m \cdot (m-1) = k$. Formally, the score for the image will be $100 \cdot a/k$, where a is the number of correct "edges". Points for the images in a test are summarized. If you do not provide an answer to the image, then the score for it is 0.

Problem A contains tests with parameters $p = 64$, $m = 8$, $k = 112$. Problem B contains tests with parameters $p = 32$, $m = 16$, $k = 480$. Problem C contains tests with parameters $p = 16$, $m = 32$, $k = 1984$. Each problem is divided into three subproblems.

Input

The archive with data is available at the link https://icpc.sgu.ru/files/shuffled-images-data.zip or https://yadi.sk/d/BmnFhkaD1Vy4vA. The data is divided into three folders: `data_train`, `data_test1_blank`, and `data_test2_blank`. Each of the folders contains three subfolders: 64, 32 and 16. Each of the subfolders contains a set of png-files with fragment sizes 64, 32 and 16 respectively. You need to process all images (or any subset of them) and send a text file containing information about the permutations of fragments for each of the files.

Files in the `data_train` folder are intended for training. This folder contains additional files `data_train_64_answers.txt`, `data_train_32_answers.txt` and `data_train_16_answers.txt` containing the correct answers for all images in corresponding subfolder. Additionally, this folder contains subfolders `64-sources`, `32-sources` and `16-sources`, which contains source images for the corresponding set.

Images in the `data_test1_blank` folder are intended for preliminary testing, images in the `data_test2_blank` folder are intended for final (system) testing.

Output

A submission is a text file with the following contents:

```
file_name_1
permutation_1
file_name_2
permutation_2
...
file_name_n
permutation_n
```

where `file_name_i` is the name of the image file for which the answer is provided, and `permutation_i` is the permutation of integers from 0 to $m^2 - 1$. Enumerate the fragments in the image from left to right, top to bottom. The i-th number in the permutation should be equal to the fragment index of the given (i.e., shuffled) image, which is located in the i-th fragment of the resulting image.

All filenames in the text file sent out must be different and correspond to the image filenames in the corresponding folder. It is allowed not to provide answers for one or more images.

Problems A1, B1 and C1 contains tests from the `data_train` folder with fragment sizes 64, 32 and 16 respectively. Answers to the training data are open to you, but you can send your answers if you want. These tasks are only needed to test your solutions. These problems will be hidden before official system testing.

Problems A2, B2 and C2 contains tests from the `data_test1_blank` folder with fragment sizes 64, 32 and 16 respectively. Your solutions will be checked and evaluated immediately after each submission. Use these problems to check and estimate the quality of your approach. These problems will be hidden before the official system testing.

Problems A3, B3 and C3 contains tests from the `data_test2_blank` folder with fragment sizes 64, 32 and 16 respectively, by which the winners will be determined (official system tests). Your solutions will not be judged during the contest, but the format will be checked. A properly formatted solution will receive 0 points during the contest. After the contest, for each participant we will use the *last* correctly formatted (0 points) submission on problems A3, B3 and C3 to evaluate the quality of the solution. **These values will be used to determine the winners**.