



Problems

- A. Compare Images
- B. Binary Image
- C. Greyscale image histogram
- D. Artist Preference Color Model
- E. Posterize
- F. Basic Text Search Engine
- G. Convolution
- H. Sum of All Even Numbers in A Range
- I. Clockwise Rotate
- J. Modulo

Advice, hints, and general information

- The problems are not sorted by difficulty.
- Your solution programs must read input from *standard input* (e.g. System.in in Java or cin in C++) and write output to *standard output* (e.g. System.out in Java or cout in C++). For further details and examples, please refer to your administrator guide and Domjudge documentation.
- For information about which compiler flags and versions are used, please refer to your administrator guide. (Python 2.7.17, Oracle Java 1.8.0_144, gcc 7.5.0 (C, C++ std14)).
- Your submissions will be run multiple times, on several different inputs. If your submission is incorrect, the error message you get will be the error exhibited on the first input on which you failed.
 - E.g., if your instance is prone to crash but also incorrect, your submission may be judged as either “Wrong Answer” or “Run Time Error”, depending on which is discovered first. The inputs for a problem will always be tested in the same order.
- If you think some problem is ambiguous or underspecified, you may ask the judges for a clarification request through the Domjudge system. The most likely response is “No comment, read problem statement”, indicating that the answer can be deduced by carefully reading the problem statement or by checking the sample test cases given in the problem, or that the answer to the question is simply irrelevant to solving the problem.
- In general, we are lenient with small formatting errors in the output, in particular whitespace errors within reason, and upper/lower case errors are often (but not always) ignored. But not printing any spaces at all (e.g. missing the space in the string “1 2” so that it becomes “12”) is typically not accepted. The safest way to get accepted is to follow the output format exactly.
- For problems with floating point output, we only require that your output is correct up to some error tolerance. For example, if the problem requires the output to be within either absolute or relative error of 10^{-4} , this means that
 - If the correct answer is 0.05, any answer between 0.0499 and .0501 will be accepted.
 - If the correct answer is 500, any answer between 499.95 and 500.05 will be accepted.
- Any reasonable format for floating point numbers is acceptable. For instance, “17.000000”, “0.17e2”, and “17” are all acceptable ways of formatting the number 17. For the definition of reasonable, please use your common sense.

Problem A

Compare Images

Time Limit: 3 seconds
Memory Limit: 512 Megabytes

Problem description

Grayscale Images are represented as a matrix with n rows and m columns where n, m is the number of pixels in each sides of the images. To determine that two images are identical or not, An Nam uses a simple method as follow:

On the matrix represented of the image, He calculates the sum of values on each pixel of the image. Then he calculates in the same way with another image to get another sum value. After that, he computes the distance between two images by taking the absolute value of the subtraction between two sum value received above. Because the size of each image could be million pixels, the sum value could excess the representation range of a 32-bit integer number. An decided to use modulo operation to compute the sum values modulo with $10^9 + 7$ before subtracting the sum values in distance calculation. Help An to write a program to identify identical images by applying above method.

Input

The first line contains four integers: n_1, m_1 and n_2, m_2 ($0 \leq n_1, m_1, n_2, m_2 < 1024$), the number rows and columns of the first and second image matrix.

The next n_1 lines contain m_1 integer numbers a_{ij} where $0 \leq a_{ij} \leq 255$ - the grey value of the pixel in the first image separated by spaces ($0 \leq i \leq n_1, 0 \leq j \leq m_1$).

The next n_2 lines contain m_2 integer numbers b_{ij} where $0 \leq b_{ij} \leq 255$ - the grey value of the pixel in the second image separated by spaces ($0 \leq i \leq n_2, 0 \leq j \leq m_2$).

Output

If both images are identical, the “identical” string will be returned, otherwise, “different” will be given. If the size of both images is not equal, they will never be considered as “identical”.

Example:

Input	Output
4 3 4 3 10 10 10 10 10 10 10 10 10	different

10 10 10 10 15 10 15 10 10 15 15 10 15 15 15	
--	--

Input	Output
2 3 3 2 10 30 30 10 140 140 10 30 10 30 140 140	different

Input	Output
3 2 3 2 10 100 10 200 100 200 10 100 10 200 100 200	identical