

Problem A. AVL Trees

Input file: av1.in
Output file: av1.out
Time limit: 5 seconds
Memory limit: 256 megabytes

AVL trees invented by Russian scientists Adelson-Velskiy and Landis are used for sorted collection data structure. The rooted binary tree is called balanced if for each vertex the height of its left subtree and the height of its right subtree differ by at most one. The balanced binary search tree is called the AVL tree.

There can be several AVL trees with the given number of vertices. For example, there are 6 AVL trees with 5 vertices, they are shown on the picture below.

Given n and h find the number of AVL trees that have n vertices and height h . Since the answer can be quite large, return the answer modulo 786 433.

Input

Input file contains n and h ($1 \leq n \leq 65\,535$, $0 \leq h \leq 15$).

Output

Output one number — the number of AVL trees with n vertices that have height h , modulo 786 433.

Example

av1.in	av1.out
7 3	16

Note

Note that 786 433 is prime, and $786\,433 = 3 \cdot 2^{18} + 1$.

Problem B. Reverse the bits

Input file: `rev.in`
Output file: `rev.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

Let $n = 2^s$, where $s \geq 0$ is integer. Consider $b_{s-1}b_{s-2} \dots b_0$ binary representation of integer i ($0 \leq i < n$) with leading zeros. Let $c_j = b_{s-1-j}$, so $c_{s-1}c_{s-2} \dots c_0$ is reversed binary representation of i . Let define a_i such number that $c_{s-1}c_{s-2} \dots c_0$ is its binary representation.

For example $s = 3, i = 3$. Binary representation of $i = 011_2$, reversed binary representation of $i = 110_2$ and $a_3 = 6$ for $s = 3$.

You should answer the queries for calculating sum $(a_l + a_{l+1} + \dots + a_r) \bmod (10^9 + 7)$

Input

In the first line of input are written integer s ($0 \leq s \leq 31$). In the second line are written integer k — number of queries ($1 \leq k \leq 5 \cdot 10^5$). The next k lines contain descriptions of queries, each consists of two integers l, r ($0 \leq l \leq r < 2^s$).

Output

Output k lines with answers for queries.

Examples

rev.in	rev.out
3 1 3 3	6
2 2 0 3 0 1	6 2
25 1 17 239	742367723

Problem C. Robots' DNA

Input file: `robots.in`
Output file: `robots.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

New inventions in DNA building technology allowed to conduct the greatest experiment in building biological robots. It was done by scientists from Research Institute of Given Strings (RIGS).

The DNA of these robots was selected to have $M = 2^n$ characters for some $n \geq 2$. This was done for simplifying the management software. Moreover, for technical reasons the Robots' DNA is not an ordinary string, but a cyclic one, so it can be read from an arbitrary position.

One interesting thing to be investigated during this experiment is the behaviour of Robots' mutations. After a long time, there have been found some different kinds of robots. To reconstruct the mutation tree, scientists from RIGS need to solve one particular task. Namely, they need to compute the similarity coefficient of two Robots' DNAs. The similarity coefficient is the number of matching letters in the best alignment of two given DNAs. The more letters match, the better is alignment.

You need to write a program that will find the best alignment.

Input

The first line of the input contains one integer M ($4 \leq M \leq 131\,072$). The next two lines are the DNAs to be investigated. Both of them consist of exactly M characters from the set 'A', 'C', 'G' and 'T'.

Output

You must output the value of the best alignment and the optimal shift. The value is the number of matching characters, and the shift is the (non-zero) number of terminating characters of second DNA to be appended to the beginning of it to get the best alignment.

Example

<code>robots.in</code>	<code>robots.out</code>
16 ACGTACGTACGTACGT CGTACGTACGTACGTC	15 1

Problem D. Triangle

Input file: `triangle.in`
Output file: `triangle.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

N points are located on the plane. Three different points are chosen randomly; all sets of three points have equal probability of being chosen. These points are then connected by line segments, and the perimeter of the resulting triangle is calculated. Note that the triangle may be degenerate.

Given the coordinates of points, find the expectation of the perimeter of the resulting triangle. The *expectation* of the perimeter is the sum $\sum_{\Delta} p_{\Delta} \cdot f_{\Delta}$ where p_{Δ} is the probability that the triangle Δ is chosen and f_{Δ} is its perimeter.

Input

The first line of input contains two integers H and W ($1 \leq H, W \leq 700$). Next H lines contain W symbols each; j -th symbol of i -th line is '1' if there is a point with coordinates (i, j) , and '0' otherwise. There are at least three points present.

Output

The first line of output should contain one real number — the expectation of triangle perimeter. Your answer should be accurate to at least six digits after the decimal point.

Examples

<code>triangle.in</code>	<code>triangle.out</code>
11 20 10000000001000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 10000000000000000000	34.142135624
3 3 101 010 101	5.794112550

Problem E. Comparing Strings

Input file: twostrings.in
Output file: twostrings.out
Time limit: 2 seconds
Memory limit: 256 mebibytes

You are given two strings S_1 and S_2 consists of uppercase english letters. For each string is chosen starting point k_j . After that all symbols of string starting from k_j are printed. After printing of the last element of the string the process continues from the first character of the string. Now we have two infinite sequences of symbols T_1 and T_2 .

For all pairs k_1, k_2 we can calculate the value $\lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n eq(T_1[i], T_2[i])}{n}$ where $T_j[i]$ — i -th symbol of string T_j and $eq(a, b)$ is 1 for equal characters and 0 for different. You are to find all possible values for this expression (average number of coincident symbols in shifted strings).

Input

In the first line of input file there is a non-empty string S_1 . In the second line of input file there is a non-empty string S_2 . The length of each string does not exceed 10^5 .

Output

Print all possible distinct values for average number of coincident symbols in the form of irreducible fractions in ascending order. Separate the numerator and denominator by “/” character.

Examples

twostrings.in	twostrings.out
CAB BCACAC	1/6 1/3 1/2
ABRACADABRA TEST	0/1
AAAB BABA	1/4 3/4