Problem A
Olympic Ranking

Time limit: 1 second
Memory limit: 1024 megabytes

Problem Description

Pac-Man is a maze-chase video game developed in 1980s. The player controls the character "Pac-Man" to eat dots in a maze while avoiding enemy characters "ghosts." All characters may move in four directions: up, down, left, right. The game ends when one of the following two conditions is met:

1. Pac-Man eats all dots in the maze. In this case, the player wins.
2. A ghost catches Pac-Man. In this case, the player loses.

Figure 1: Pac-Man gameplay (image from Wikipedia)

Adam is learning how to create games with modern programming tools. To practice the skills, he tries to reproduce the Pac-Man game with some modification. In Adam's game, the playable character is a "ghost," and the enemy character is "Pac-Man." Since he changes the roles of the ghost and Pac-Man, he also changes the ending conditions of the game.

1. Pac-Man eats all dots in the maze. In this case, the player loses.
2. The ghost controlled by the player catches Pac-Man. In this case, the player wins.

Adam has almost developed the first full functioning version of his game. He wants to test his game and creates a simple stage for testing. The maze of the stage is based on a 10-by-10 grid. We label the cell at the intersection of row r and column c with (r, c). In this problem, rows and columns are numbered from 0 to 9. Each grid cell contains exact one dot. The exterior boundary of the grid are walls. No characters may move to the area outside of the grid. Inside the grid, there are no walls or obstacles. All characters may move freely from a cell to any cell adjacent to it. Note that two grid cells (r_1, c_1) and (r_2, c_2) are adjacent to each other if and only if $|r_1 - r_2| + |c_1 - c_2| = 1$.

Adam has to prepare the movements of Pac-Man for the testing. He needs a set of Pac-Man's

trajectories with diversity, but any trajectory must satisfy the following requirements.

- Pac-Man eats all dots in the maze if it follows the trajectory.
- Pac-Man moves at most 10000 steps.

Adam needs your help to generate a trajectory starting at cell (x, y). Please write a program to generate a trajectory of Pac-Man satisfying all requirements above and starting at cell (x, y).

Input Format

The input has exactly one line which consists of two space-separated integers x and y. You are asked to generate a trajectory starting at cell (x, y).

Output Format

You must output a requested trajectory in the following format:

The trajectory is represented by $m + 1$ lines where m is the number of steps of the trajectory. The i-th line contains two integers r_i and c_i separated by exactly one space. The trajectory starts at the cell (r_1, c_1), and Pac-Man will be in cell (r_i, c_i) after moving $i - 1$ steps along the trajectory for $1 < i \leq m + 1$.

Technical Specification

- $m \leq 10000$
- $x, y, r_i, c_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ for all $i \in \{1, 2, \ldots, m + 1\}$.
- Cells (r_i, c_i) and (r_{i+1}, c_{i+1}) are adjacent to each other for all $i \in \{1, 2, \ldots, m\}$.
- $\{(r_1, c_1)\} \cup \{(r_2, c_2)\} \cup \cdots \cup \{(r_{m+1}, c_{m+1})\} = \{(r, c) : r \in \{0, 1, \ldots, 9\}, c \in \{0, 1, \ldots, 9\}\}$
- If there are multiple solutions, then you may output any of them.

Sample Input 1

```
0  0
```

Sample Output 1

```
0  0
0  1
0  2
. . .
. . .
. . .
9  2
9  1
9  0
```

Note

The sample output section does not contain the correct output, since it ignores a large part of the answer. Please download the correct sample test cases from the judge system.

Problem B
Perfect Number

Time limit: 2 seconds

Memory limit: 1024 megabytes

Problem Description

There is a transparent tape. Its length is exact one meter (10^9 nanometers). In this problem, all numbers are integers, and we use a number to denote a position on the tape. The number p denote the position of the point has a distance p nanometers from the head of the tape.

Bob is a master dyer, so he can color the tape precisely in nanometer scale. He colors two sectors $[p_1, q_1]$ and $[p_2, q_2]$ into red. The color of the tape within the range between p_1 and q_1 is red. The color of the tape within the range between p_2 and q_2 is also red. And the rest parts of the tape remain transparent.

To verify Bob's skill, we ask Ben, the tape folding master, to help us. Ben can fold the tape perfectly at any position. If Ben fold the tape at x, then the new position of a certain point p will be one of the following cases.

- If $p = x$, then it becomes the new head of the tape, i.e, it becomes 0.
- If $p > x$, then it becomes $p - x$.
- If $p < x$, then it becomes $x - p$.

After Ben folds the tape, we measure the total length of the red part of the new tape. If the red part has the expected length, then we will believe Bob and Ben are both masters in their skills. Obviously, the color of some position of the new tape is determined by the colors of the corresponding positions of the old tape. A position of the new tape is colored in red if one of the corresponding positions in the old tape is colored in red.

Bob has already colored the tape, and Ben has proposed the positions to be folded. Please write a program to compute the expected lengths colored in red.

Input Format

The first line contains four space-separated integers p_1, q_1, p_2 and q_2. Bob has colored the sectors $[p_1, q_1]$ and $[p_2, q_2]$. The second line contains an integer q indicating that Ben has made q proposals. Each of the remaining q lines contains an integer x indicating the positions to be folded by Ben. Note that the q proposals are independent to each other. There is only one folding point in one proposal.

Output Format

For each position, output the expected total length of the new tape that are colored in red.

Technical Specification

- $0 \le p_1 < q_1 < p_2 < q_2 \le 10^9$
- $0 \le x \le 10^9$
- $q \le 10^6$

Sample Input 1

```
1  3  8  9
10
1
2
3
4
5
6
7
8
9
10
```

Sample Output 1

```
3
2
3
3
2
3
3
3
3
3
```

Problem C
Return Home

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

There are n magical circles on a plane. They are centered at $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, respectively. In the beginning, the radius of each circle is 0, and the radii of all magical circles will grow at the same rate. When a magical circle touches another, then it stops growing. Write a program to calculate the total area of all magical circles at the end of growing.

Input Format

The first line contains an integer n to indicate the number of magical circles. The i-th of the following n lines contains two space-separated integers x_i and y_i indicating that the i-th magical circle is centered at (x_i, y_i).

Output Format

Output the total area of the circles.

Technical Specification

- $2 \le n \le 2000$
- $x_i, y_i \in [-10^9, 10^9]$ for $i \in \{1, 2, \ldots, n\}$.
- All (x_i, y_i)'s are disinct points.
- A relative error of 10^{-6} is acceptable.

Sample Input 1

```
4
0 0
1 0
1 1
0 1
```

Sample Output 1

```
3.14159265359
```

Sample Input 2

```
3
0 0
0 1
2 0
```

Sample Output 2

```
8.639379797371932
```

Almost blank page

Problem D
Drunk Passenger

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

David is a farmer and has a large farm. The shape of the farm is a square. A sqaure is a quadrilaterial that has four equal sides and four equal angles. The length of any side of David's farm is one kilometer, so the area of his farm is slightly greater than the total area of 140 standard football fields.

David is old and very ill. He feels that his days are numbered. Worrying that his spouse Dora and their three children, Alice, Bob, and Cliff, will have a dispute over the ownership of the farm after he passes away, he plans to divide the farm into four parts, and then to allocate each part to one of his family members. He decides to write his last will as follows.

1. Assume that the shape of the farm is a sqaure $ABCD$ where $A = (0,0)$, $B = (1,0)$, $C = (1,1)$, $D = (0,1)$.
2. Let $E = (0.5, 0)$, $F = (1, 0.5)$, $G = (0.5, 1)$, $H = (0, 0.5)$ be the midpoints of $\overline{AB}$, $\overline{BC}$, $\overline{CD}$, $\overline{DA}$, respectively.
3. Let $area(PRQS)$ to denote the area of the quadrilaterial $PQRS$.
4. Please find a point X *strictly inside* the square $ABCD$ such that

$$area(AEXH) : area(BFXE) : area(CGXF) = p : q : r$$

   Note that X cannot be on the boundary of the square $ABCD$.
5. Allocate the land in $AEXH$, $BFXE$, $CGXF$, $DHXG$ to Alice, Bob, Cliff and Dora, respectively.

David is still adjusting the numbers p, q, r, and his lawyer, Reed, has to read David's last will carefully. Reed finds that it is impossible to find such point X if David gives an improper set of the numbers such as $p = 1, q = 2, r = 1$. However, there are proper sets of the numbers p, q, r that allow us to find the point X satisfying David's last will. For instance, let $p : q : r = 2 : 3 : 2$, the following figure shows a possible position of X.
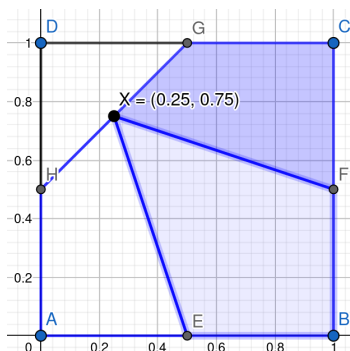

Figure 2: $area(AEXH) : area(BFXE) : area(CGXF) = 2 : 3 : 2$

Please write a program to help Reed to determine whether it is possible to find a point X satisfying David's last will for a given set of numbers p, q, r. If possible, please output one possible position of X to Reed.

Input Format

The input contains one line only. The line contains three space-separated positive integers p, q, r.

Output Format

If there does not exist a point X satisfying David's last will, then output -1. Otherwise, output two irreducible fractions x and y such that (x, y) can be the point X satisfying David's last will. You must output an irreducible fraction $t = \frac{n}{d}$ as n/d with a positive denominator and use exactly one space to separate x and y.

Note: the numerator and the denominator of any irreducible fraction are integers and do not have common divisors other than 1 and −1.

Technical Specification

- $p, q, r \in \{1, 2, \ldots, 10^6\}$

Sample Input 1

```
1  1  1
```

Sample Output 1

```
1/2  1/2
```

Sample Input 2

```
1  2  1
```

Sample Output 2

```
-1
```

Sample Input 3

```
2  3  2
```

Sample Output 3

```
1/4  3/4
```

Problem E
Mountain Park

Time limit: 5 seconds
Memory limit: 1024 megabytes

Problem Description

A binary string is a string consisting of only 0's and 1's. Elsa, Eric's boss, gave him a binary string s of length 20 and asked him to modify s into another binary string t within D days.

Eric really hates this task and therefore never modifies more than one character in a day. However, being forced to show Elsa the daily progress, Eric must modify some characters of the string every day. That means, the only possible way for Eric is to modify exact one character in each day before he finishes the task.

It is obviously cheating to have a character changed to something other than 0 and 1. Moreover, Eric will be caught cheating if the string is modified into the same binary string twice since Elsa has a good memory. That is, before the string is modified into t, all modifications result in unique strings. Note that Eric cannot modify the string into s which is the string given by Elsa, either.

Eric wants to spend as much time as possible. He is wondering if he can spend exact D days to have the string s modified into t. Please write a program to help Eric.

Input Format

The input contains three lines. The first line contains a binary string s. The second line contains a binary string t. The third line contains an integer D. Elsa asked Eric to modify the binary string s into t within D days.

Output Format

If there is no way to achieve what Eric wants, output -1. Otherwise, output D lines to represent one possible way. The i-th line contains a binary string, the result of the modification on the i-th day.

Technical Specification

- The strings s and t consist of only 0's and 1's.
- The length of s and the length of t are both 20.
- $1 \leq D \leq 500000$
- If there are multiple solutions, then you may output any of them.

Sample Input 1

```
00000000000000000000
11111111111111111111
5
```

Sample Output 1

```
-1
```

Sample Input 2

```
00000000001111111111
10000000001111111111
5
```

Sample Output 2

```
00000000001111111110
00000001001111111110
10000001001111111110
10000000001111111110
10000000001111111111
```

Problem F
F

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

There are n children (numbered from 1 to n) learning the arithmetic operations, which include *addition* "+", *subtraction* "−", *multiplication* "×", and *division* "÷" on rational numbers.

In the beginning, each child has a paper sheet with only a zero on it. Their teacher, Frank, will then give them q operations. The i-th operation consists of an operator c_i and an integer x_i. The children numbered $\ell_i, \ell_i + 1, \ldots, r_i$ have to append the operator c_i and the integer x_i to their paper sheets. After that, every child has an expression on their sheet to be evaluated.

For example, suppose that $n = 3$, $q = 2$, c_1 is "+", $x_1 = 1$, $\ell_1 = 1$, $r_1 = 2$, c_2 is "−", $x_2 = 2$, $\ell_2 = 2$, and $r_2 = 3$. The expressions on the sheets are are $0 + 1$, $0 + 1 - 2$ and $0 - 2$ for children 1, 2 and 3, respectively.

Since Frank is really lazy and wants to verify the answers quickly, he asks you to calculate the sums of the values of all children's expressions. Suppose that the value of the expression assigned to child i is $\frac{a_i}{b_i}$, then the value will be $a \times b^{-1} \bmod 10^9 + 7$ instead, where b^{-1} denotes the integer satisfying $b \times b^{-1} \equiv 1 \mod 10^9 + 7$. If the sum is not in $[0, 10^9 + 7)$, then the sum modulo $10^9 + 7$ should be returned to Frank.

Note: The arithmetic operations has PEMDAS rule, that is, multiplications and divisions should be evaluated before evaluating additions and subtraction.

Input Format

The first line consists of two space-separated integers n and q. The i-th of the following q lines consists of four space-separated tokens ℓ_i, r_i, c_i, x_i. For the sake of convenience, * and / are used to represent the multiplication and the division operators, respectively.

Output Format

Output the number that you should return to Frank.

Technical Specification

- $1 \le n \le 10^5$
- $1 \le q \le 10^5$
- $\ell_i, r_i \in [1, n]$ for all $1 \le i \le q$.
- $c_i \in \{+, -, *, /\}$ for all $1 \le i \le q$.
- For all $1 \le i \le q$, $x_i = 0$ implies that c_i is not /.
- $x_i \in [0, 10^9 + 7)$ for all $1 \le i \le q$.

Sample Input 1

```
3 2
1 2 + 1
2 3 - 2
```

Sample Output 1

```
1000000005
```

Problem G
G

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

There is a rectangle garden in front of Gina's house. The garden can be seen as an n-by-m rectangular grid. All grid cells are identical squares, and two grid cells are considered adjacent if they share an common edge.

Gina loves cacti and wants to plant as many cacti as possible in the garden. However, there are some constraints on planting cacti.

- The soil can be too wet in some of the cells and therefore is not suitable for cacti. Gina cannot plant cacti in those cells.

- Since the soil in each cell is not fertile enough to grow two or more cacti, Gina may plant at most one cactus in a cell.

- At most one cactus can be planted in any pair of adjacent cells. Otherwise, the cacti in those cells may be harmed by their neighbor's thorns.

Please write a program to help Gina calculate the maximum possible number of cacti can be planted, and also a way of plantings that meet the listed constraints.

Input Format

The first line contains two space-separated integers n and m meaning the garden is an n-by-m grid. Then, each of the following n lines contains a string of m characters. These characters are either '.' or '*'. The j-th character of the i-th of these lines indicates whether the soil in the grid cell on the i-th row and the j-th column is suitable for planting a cactus. '.' means it is suitable, and '*' means it is not suitable.

Output Format

First, output the maximum possible number of cacti on the first line. Then, output n lines, each line containing a string of m characters. Each of the characters must be one of '.', '*' and 'C'. The j-th character of the i-th of these lines indicates the status of the grid cell on the i-th row and the j-th column. A 'C' means a cactus should be planted in that particular cell, and the other cells should be identical to the corresponding position of the input.

Technical Specification

- $1 \leq nm \leq 10^5$
- If there are more than one possible way of planting, any of them will be accepted.

Sample Input 1

```
3 3
*.*
...
*.*
```

Sample Output 1

```
4
*C*
C.C
*C*
```

Sample Input 2

```
2 4
*..*
....
```

Sample Output 2

```
3
*C.*
C.C.
```

Problem H
H

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

> There's no such thing as public opinion.
>
> ———————————————
>
> *Jordan Ellenberg, American*
> *Mathematician*

In K City lives n residents who want to build a connection network with each other. However, some residents want the network wires colored black while the others want the wires colored white. The opinion of resident i can be quantified as a number a_i. If we build a network wire between residents i and j, the cost of this wire will be $a_i \times a_j$.

The mayor of K City wants to build a network such that:

1. There are exactly $n - 1$ wires used.
2. For any two different residents i and j, there exists a sequence $p_1, \cdots, p_k$ such that $p_1 = i$, $p_k = j$ and residents p_ℓ and $p_{\ell+1}$ share a wire for $1 \le \ell < k$.

In other words, the network should be a tree.

You, the renowned mathematician of K City, want to know not only the *minimum* cost to build the network. In the name of confusion, you also want to know the *maximum* cost!

Input Format

The first line begins with a number n indicating the number of residents. The second line contains n numbers $a_1, a_2, \ldots, a_n$. The opinion of resident i is the quantified as a_i.

Output Format

Output two numbers separated by a blank in a line. The numbers are the *minimum* cost and the *maximum* cost to build the network, respectively. Since the absolute value of the costs may be extremely large, you have to modulo the answer with $10^9 + 7$. Please note that the modulo of a number (defined by Donald Knuth) is $a \bmod b = a - b\lfloor \frac{a}{b} \rfloor$. The output number should be non-negetive.

Technical Specification

- $1 \le n \le 10^6$
- $|a_i| \le 10^6$

Sample Input 1

```
10
-5 -10 -7 -7 -3 -1 -7 -5 -8 -6
```

Sample Output 1

```
58 490
```

Sample Input 2

```
10
-5 1 2 -2 -1 1 -5 5 -10 6
```

Sample Output 2

```
999999779 183
```

Sample Input 3

```
10
0 0 0 0 0 0 0 0 0 0
```

Sample Output 3

```
0 0
```

Sample Input 4

```
10
10 8 9 3 8 8 0 5 3 10
```

Sample Output 4

```
0 540
```

Problem I
I

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

Teams from variaous universities compete in ICPC regional contests for tickets to the ICPC World Finals. The number of tickets allocated to every regional contest may be different. The allocation method in our super region, Asia Pacific, is based on a parameter called site score.

Site scores will only count teams and universities solving at least one problem, in the regional contest or its preliminary contest TOPC. In 2020, the formula for calculating the site score of the Taipei-Hsinchu regional contest is much simpler than past years. Let

- U_R be the number of universities solving at least one problem in the regional contest.
- T_R be the number of teams solving at least one problem in the regional contest.
- U_O be the number of universities solving at least one problem in TOPC.
- T_O be the number of teams solving at least one problem in TOPC.

The site score of 2020 Taipei-Hsinchu regional contest will be $56U_R + 24T_R + 14U_O + 6T_O$. Please write a program to compute the site score of the 2020 Taipei-Hsinchu regional contest.

Input Format

The input has only one line containing four blank-separated positive integers U_R, T_R, U_O, and T_O.

Output Format

Output the site score of the 2020 Taipei-Hsinchu regional contest.

Technical Specification

- $0 < U_R \leq T_R \leq 120$
- $0 < U_O \leq T_O \leq 1000$

Sample Input 1

```
1 1 1 1
```

Sample Output 1

```
100
```

Sample Input 2

```
1 10 100 1000
```

Sample Output 2

```
7696
```

Note

The problem statement is fiction. The real site score has a different formula.

Almost blank page

Problem J
J

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

Alex is attending the first edition of Robotic World Championship of Table Tennis. A competition that have all of the matches having the same rules listed below:

- A match shall consist of the best of 7 games, i.e., the results of matches must be 4 games to k, where $0 \le k \le 3$.
- A game shall be won by the player first scoring 11 points unless both players score 10 points, when the game shall be won by the first player subsequently gaining a lead of 2 points. For example, a game can be won at scores like 11-5, 11-9 or 12-10, but not 10-5 or 11-10.
- After each 2 points have been scored the receiving player shall become the serving player and so on until the end of the game, unless both players score 10 points, when the sequences of serving and receiving shall be the same but each player shall serve for only 1 point in turn. That is, the servicing order of the first 20 points is `AABBAABBAABBAABBAABB`, and will be followed by `ABABAB`. . . if necessary.
- The player serving first in a game shall receive first in the next game of the match.

Experience tells that when two robots clashes into each other, the variances affecting their winning chances can be simplified to who's serving for the point. This is due to the performances of the robots are physically consistent and won't be affected mentally.

Alex have listed some of the possible matchups, simplified to the winning chance of each servicing point of the robots, for you. Now it is your job to help him calculate the winning chance of each match for them.

Input Format

The first line of the input consists of a single number T, indicating that there will be T test cases following.

Each of the following test case consists of two space-separated real numbers P_A and P_B in one line, where P_A denotes the Robot A's chance of winning the point when A is serving and P_B denotes the Robot B's chance of winning the point when B is serving.

The Robot A always serves first in the very first game of the match.

Output Format

For each test case, output one real number in one line: the winning chance of A.

Technical Specification

- $T \leq 100$
- $0 \leq P_A \leq 1$ and has at most 2 digits after the decimal point in the input.
- $0 \leq P_B \leq 1$ and has at most 2 digits after the decimal point in the input.
- $0 < P_A + P_B < 2$
- The answer will be considered correct if it is within an absolute error of 10^{-8} of the correct answer.

Sample Input 1

```
3
1 0
0.5 0.5
0.00 1.00
```

Sample Output 1

```
1
0.5
0.000000000
```

References

The rules are revised from the Chapter 2 "The Laws of Table Tennis" of The International Table Tennis Federation (ITTF) Handbook 2020.