ANZAC CONTEST

MARCH 21, 2020

Contest Problems

A: Exam
B: Goat Rope
C: Repeating Goldbachs
D: Illiteracy
E: Knockout
F: Rectangles
G: Random Index Vectors
H: Area Rug
I : Sculpture
J : Time Limits
K: To Tell the Truth

Problem set contains 11 problems over 24 pages

This page is intentionally left (almost) blank.

Problem A
Exam

You and your friend have just taken a True/False exam. Your friend has been to see the instructor, so they know how many answers they got right (but not which ones). You compare notes: you know your answers and your friend's answers. What is the maximum number of answers you could have gotten right?

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. Each test case will begin with a line containing a single integer n ($0 \leq n \leq 1000$), which is the number of answers your friend got right on the exam. Each of the next two lines will contain a string s ($\max[n, 1] \leq |s| \leq 1000$, $s \in \{T, F\}^*$). The two strings will be of the same length. The first line represents your answers; the second line represents your friend's answers. The order of answers is the same in both strings: the first letter is the answer to question 1, the second to question 2, and so on.

Output

Output a single integer, which is the maximum number of answers you could have gotten right.

Sample Input 1	Sample Output 1
3 FTFFF TFTTT	2

Sample Input 2	Sample Output 2
6 TTFTFFTFTF TTTTFFTTTT	9

This page is intentionally left (almost) blank.

Problem B
Goat Rope

You have a fence post located at (x, y) and a goat. You also have a house, which you model as an axis-aligned rectangle with opposite corners at (x1, y1) and (x2, y2). You want to give the goat as much room to roam as possible, but you don't want the goat to be able to touch the house. As a guide to how much rope you should buy, determine the minimum distance from the post to your house.

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. Each test case will consist of a single line containing six space-separated integers, all in the same units:

$$x\ y\ x1\ y_1\ x_2\ y_2$$

where (x, y) is the location of the post, and (x_1, y_1) and (x_2, y_2) are opposite corners of the house. The following are guaranteed:

- All values are between -1000 and 1000 inclusive.
- $x_1 < x_2$ and $y_1 < y_2$.
- The post is not inside the house or on the border.
- At least one of these is true: $x < x_1$ or $x > x_2$ or $y < y_1$ or $y > y_2$.

Output

Output a single real number, which is the minimum distance from the post to your house (in the same units as the inputs). Output this number rounded to exactly 3 decimal places.

Sample Input 1	Sample Output 1
7 4 0 0 5 4	2.000

Sample Input 2	Sample Output 2
6 0 0 2 7 6	2.000

Sample Input 3	Sample Output 3
4 8 7 8 9 9	3.000

This page is intentionally left (almost) blank.

Problem C
Exam

The Goldbach Conjecture states that any even number greater than 3 can be expressed as the sum of two primes (primes are numbers that have exactly two factors: themselves and 1). It has never been proven for all even numbers, but it has been demonstrated to be true for all of the numbers that we'll use for this problem. Consider any even integer $x > 3$. There may be many pairs of primes which sum to x. Take the pair with the largest difference. That difference must be even, and less than x. So, repeat the trick. How many steps does it take until you reach an even number less than 3 (2 or 0)?

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. Each test case will consist of a single line with a single integer x ($0 \leq x \leq 10^6$, x is even).

Output

Output a single integer, which is the count of Repeating Goldbach steps until the number is less than 3.

Sample Input 1	Sample Output 1
20	3

Sample Input 2	Sample Output 2
30	4

Sample Input 3	Sample Output 3
40	5

Sample Input 4	Sample Output 4
50	6

Sample Input 5	Sample Output 5
60	7

Sample Input 6	Sample Output 6
70	8

This page is intentionally left (almost) blank.

Problem D
Exam

Illiteracy is a simple puzzle game. After the contest, if you'd like to play it, you can find it here: https://le-slo.itch.io/illiteracy. Of course, during the contest, it won't be accessible (and you've got better things to do!) The game has a string of 8 icons. The icons in the game are very artistic, but for simplicity, we'll just call them $A, \ldots, F$. Clicking any icon has a unique effect on the other icons. Most of the icons *Rotate* other icons. That means that they change $A \Rightarrow B$, $B \Rightarrow C$, $C \Rightarrow D$, $D \Rightarrow E$, $E \Rightarrow F$, and $F \Rightarrow A$. There are 8 icon positions in a row, numbered left to right, 1 to 8. Here's what each of the icons do when clicked:

- A: *Rotates* the icon immediately to the left, if there is one, and immediately to the right, if there is one.
- B: If not on the end, changes the icon immediately to the right to be same as the one immediately to the left (does nothing on the ends). This is the only icon that doesn't *Rotate* other icons.
- C: *Rotates* the mirror image (when clicked in position x, *Rotates* $9 - x$. e.g. clicking 1 *Rotates* 8, 2 *Rotates* 7, etc.)
- D: *Rotates* all of the icons between this one and the closest end. (e.g. clicking 3 *Rotates* 1 and 2, 5 *Rotates* 6, 7 and 8. Clicking this icon on the end does nothing.)
- E: *Rotates* the closest end, and also the position which is the same distance in the opposite direction. (e.g. clicking 1 does nothing, 2 *Rotates* 1 and 3, 3 *Rotates* 1 and 5, 5 *Rotates* 8 and 2, 7 *Rotates* 8 and 6, etc.)
- F: *Rotates* another position with this pattern: Clicking 1 *Rotates* 5, 2 *Rotates* 1, 3 *Rotates* 6, 4 *Rotates* 2, 5 *Rotates* 7, 6 *Rotates* 3, 7 *Rotates* 8, and 8 *Rotates* 4. In other words, clicking an icon in position x *Rotates* $(x + 9)/2$ if x is odd, and *Rotates* $x/2$ if x is even.

Given a starting and target configuration, what's the smallest number of steps needed to get from the start to the target?

Input

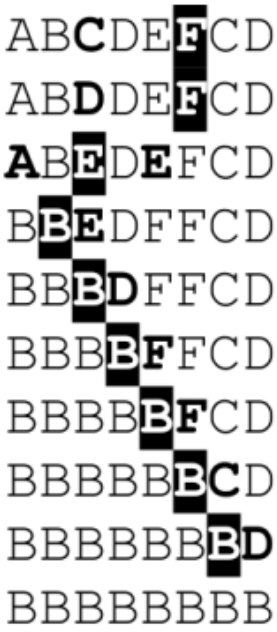
Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. Each test case will consist of exactly two lines. Each line will have a string of length exactly 8, consisting only of the upper-case letters A, B, C, D, E and/or F. The first line holds the starting position, the second holds the target.

Output

Output a single integer, which is the smallest number of steps needed to get from the start to the target, or -1 if it isn't possible.

Explanation

Here is one possible way to solve the first case of the sample data in the minimum 9 steps. The icon clicked is in Inverse, the icons which change are in *Bold*.



Sample Input 1	Sample Output 1
ABCDEFCD BBBBBBBB	9

Sample Input 2	Sample Output 2
BBBBBBBB ABCDEFCD	-1

Problem E
Knockout

The solitaire game Knockout is played as follows. The digits from 1 to 9 are written down in ascending order. In each turn, you throw a pair of six-sided dice; you sum the dice, and cross out some set of digits, of your choice, that sum to the same total. If you cannot, the game ends and your score is the remaining digits, taken as a single number. Otherwise, you throw the dice again and continue.

This game can be played to either minimize or maximize your score. Given a position of the game (what digits remain) and a roll of the dice, determine which digits you should remove and what your expected score would be for both versions of the game, assuming you make the best moves possible for whichever version you're playing for the remainder of the game. The expected score is the sum of all possible scores weighted by their probabilities (presuming optimum play).

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs.

Each test case will consist of a single line containing a string of digits d ($1 \leq |d| \leq 9$) and two integers a and b ($1 \leq a, b \leq 6$), all separated by spaces. The string of digits d will contain a subset of the digits $1, \ldots, 9$ in ascending order, with no digit appearing more than once. This is the current state of the game. The integers a and b represent your current throw of the dice.

Output

Output two lines, each with two parts. First, output the digits that you eliminate with your throw of the dice, as a string of digits in ascending order. If you cannot eliminate any digits, output -1. Then, output the expected score as a real number rounded to five decimal places. Output a space between the parts.

The first line represents the best result when minimizing your score and the second line represents the best result when maximizing your score. Note that it is impossible for two different combinations of digits to yield the same expected score.

Sample Input 1	Sample Output 1
1345 1 1	-1 1345.00000 -1 1345.00000

Sample Input 2	Sample Output 2
12349 3 1	13 151.70370 4 401.24546

This page is intentionally left (almost) blank.

Problem F
Rectangles

You are working on a new graphics system, which has added a new feature. Whenever you draw a figure, all the pixels in that figure flip from white to black, or from black to white. This image is what happens when three overlapping rectangles are drawn on a white field:

Starting with a white field, given a series of axis-aligned rectangles, how many pixels end up black?

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs.

Each test case will begin with a line with a single integer n ($1 \leq n \leq 100000$) indicating the number of rectangles.

Each of the next n lines will have four space-separated integers x_1, y_1, x_2 and y_2 ($0 \leq x_1 < x_2 \leq 10^9$, $0 \leq y_1 < y_2 \leq 10^9$) which represent opposite corners of a rectangle.

The rectangle consists of all pixels (x, y) such that $x_1 \leq x < x_2$ and $y_1 \leq y < y_2$, so the area of the rectangle is $(x_2 - x_1) \times (y_2 - y_1)$ pixels.

Output

Output a single integer, which is the number of pixels that are black after all of the rectangles are drawn on a white field.

Sample Input 1

```
2
0 0 4 4
1 1 3 3
```

Sample Output 1

```
12
```

Sample Input 2

```
4
0 0 10 10
1 1 11 11
2 2 12 12
3 3 13 13
```

Sample Output 2

```
72
```

This page is intentionally left (almost) blank.

Problem G
Random Index Vectors

Random Index Vectors (RIVs) are a relatively new technique for pattern matching. A RIV is a large, sparse vector of 1s and -1s. If randomly generated, the dot product of two RIVs is zero or very nearly zero, so they are orthogonal or very nearly orthogonal. They are used by assigning a RIV to various attributes, and then combining them in specific ways to form new vectors for patterns with those attributes. Then, the cosine of the angle between vectors for two patterns can be used as a measure of similarity between the patterns.

There are three basic operations on RIVs:

- Two RIVs can be ADDED element by element: $C_i = A_i + B_i$. RIVs can only have 1 and -1 as nonzero elements, so 1+1=1 and -1+-1=-1
- Two RIVs can be MULTIPLIED element by element: $C_i = A_i \times B_i$.
- One RIV can be ROTATED by some integer k, where all of the values are shifted to the left by k (towards the lower indices), and values at the start of the vector go to the end.

Because they are large and sparse, RIVs usually use a condensed representation. The vector is represented by a sorted list of indices (starting at 1) where there are non-zero values, with the index being negative if the value is -1. The representation starts with the number of non-zero indices.

For example, consider this RIV:

$$1\,0\ -1\,0\,0\,0\ -1\,0\,0\,1\,0$$

There are 4 non-zero elements at indices 1, 3, 7 and 10. In condensed representation:

$$4\,1\ -3\ -7\,10$$

Given two RIVs in condensed representation, add them, multiply them, and rotate them both. Give the resulting vectors in condensed form.

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs.

Each test case will begin with a line with two space-separated integers n ($1 \leq n \leq 10^{18}$) and k ($1 \leq k \leq n$), where n is the maximum index of the vectors and k is the number of spots to rotate by.

Each of the next two lines will have a vector in condensed form, starting with an integer m ($0 \leq m \leq 1000$) followed by m indices i ($1 \leq |i| \leq n$), all separated by spaces.

Output

Output four Random Index Vectors, one per line, in condensed form.

- On the first line, output the sum of the input vectors.
- On the second line, output the product of the input vectors.
- On the third line, output the first input vector rotated by k.
- On the fourth line, output the second input vector rotated by k.

Sample Input 1	Sample Output 1
30 13 6 6 -9 -13 18 22 26 8 -1 3 7 11 13 19 20 -27	12 -1 3 6 7 -9 11 18 19 20 22 26 -27 1 -13 6 5 9 13 23 -26 -30 8 6 7 -14 -18 20 24 28 30

Sample Input 2	Sample Output 2
20 4 9 -2 -4 -8 -11 -12 15 18 19 20 7 4 5 -10 11 15 18 -20	8 -2 5 -8 -10 -12 15 18 19 5 -4 -11 15 18 -20 9 -4 -7 -8 11 14 15 16 -18 -20 7 1 -6 7 11 14 -16 20

Problem H
Area Rug

The main room of your home is square, $n \times n$ meters. Unfortunately, the floor is dirty. You're a college student, so you hate to clean! Rather than clean it, you buy an area rug $s \times s$ meters square to cover some of the dirty spots.

Consider all of the ways that you could place the $s \times s$ area rug in the $n \times n$ room so that all $s \times s$ square feet of it cover part of the floor, axis aligned (no rotation). How many ways are there to cover a certain number of dirty spots?

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs.

Each test case will begin with a line with two space-separated integers n ($1 \leq n \leq 1000$) and s ($1 \leq s \leq \min[n, 100]$), where n is the size of one side of the room, and s is the size of one side of the new area rug.

Each of the next n lines will have a string of exactly n characters, consisting only of 'C' (a clean spot on the floor) or 'D' (a dirty spot on the floor).

Output

For each count of dirty floor spots covered, from 0 to s^2 , if the number of ways of covering that many dirty spots with an area rug of size $s \times s$ is greater than 0, output the number of spots and the number of ways of covering them on a line, separated by a space. Output them in order, smallest number of dirty spots to largest.

Explanation

In this example, there are 4 ways to cover 9 dirty spots (the corners), 16 ways to cover 5 dirty spots (the non-corner edges), and 16 ways to cover 0 dirty spots (the interior).

Sample Input 1	Sample Output 1
10 5 DDDDDDDDDD DCCCCCCCCD DCCCCCCCCD DCCCCCCCCD DCCCCCCCCD DCCCCCCCCD DCCCCCCCCD DCCCCCCCCD DCCCCCCCCD DDDDDDDDDD	0 16 5 16 9 4

This page is intentionally left (almost) blank.

Problem I
Sculpture

A modern artist has created a large outdoor sculpture. It consists of a rectangular grid, where each square cell of the grid is raised to a different height. Because the sculpture is to be placed outdoors, the artist is worried about rainwater pooling in the lower cells. He needs to figure out which cells need to have drains installed. A cell will need a drain if the four cells above, below, left and right are all higher. A cell on the edge or corner of the grid will never need a drain.

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs.

Each test case will begin with a line with two space-separated integers r and c ($1 \leq r, c \leq 100$), which are the dimensions of the artist's grid.

Each of the next r rows will contain c space-separated integers h ($0 \leq h \leq 1000$), which are the heights of the cells in centimeters. It is guaranteed that every cell's height will be different from those immediately above, below, to the left, and to the right.

Output

Output r lines with c space-separated integers each. Output a 1 if that cell needs a drain, 0 if it doesn't.

Sample Input 1

```
7 10
4 5 6 5 7 8 9 3 2 4
9 1 3 4 6 2 0 5 6 1
7 8 1 3 8 7 5 4 3 2
9 3 4 2 7 6 8 2 8 7
8 5 3 6 4 0 7 9 6 3
3 8 4 8 3 7 4 2 5 7
9 3 0 4 5 6 7 8 4 3
```

Sample Output 1

```
0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 1 0 0 0
0 0 1 0 0 0 0 0 0 0
0 1 0 1 0 0 0 1 0 0
0 0 1 0 0 1 0 0 0 0
0 0 0 0 1 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0
```

This page is intentionally left (almost) blank.

Problem J
Time Limits

Your Chief Judge needs help! He needs to set the time limit for a problem in the problem set. He has n solutions written by his judges. He knows how long each runs in the contest environment, in milliseconds. He wants to set the time limit to be at least s times the slowest solution from his judges, but as small as possible, and he wants it to be an integral number of seconds. Can you help him?

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs.

Each test case will begin with a line containing two space-separated integers n ($1 \leq n \leq 100$) and s ($1 \leq s \leq 20$), where n is the number of solutions from judges, and s is the multiplying factor. The next line will contain n space-separated integers m ($1 \leq m \leq 2,000$), which are the number of milliseconds it takes for some judge's solution to run in the contest environment.

Output

Output a single integer, which is the time limit to set for this problem. It should be in seconds, and the smallest time that is at least s times the slowest judge's solution.

Sample Input 1	Sample Output 1
2 5 200 250	2

Sample Input 2	Sample Output 2
3 4 47 1032 1107	5

This page is intentionally left (almost) blank.

Problem K
To Tell the Truth

There are n people in a room, each of whom always tells the truth, or always lies. Each of them makes a statement of the form: "Some number between a and b (inclusive) of us are telling the truth." What is the maximum number of truth-tellers in the room?

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs.

Each test case will begin with a line containing an integer n ($1 \leq n \leq 1000$) which is the number of people in the room.

Each of the next n lines will have two space-separated integers, a and b ($0 \leq a \leq b \leq n$).

Each line represents the statement of one person that "Some number between a and b (inclusive) of us are telling the truth."

Output

Output a single integer, which is the largest possible number of truth-tellers, or -1 if the statements are inconsistent.

Sample Input 1

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

Sample Output 1

```
2
```

Sample Input 2

```
8
0 1
1 7
4 8
3 7
1 2
4 5
3 7
1 8
```

Sample Output 2

```
-1
```

This page is intentionally left (almost) blank.