1. BoardGamesMarks: 30

Problem Description

Ankitha enjoys finding new games. One day, she found a grid with dimensions M*N and decided to make up a special game to play on it. When Ankitha came up with the idea for the new game, her friend Akhil joined her. She then decided to share and explain the game to him.

Akhil is given a grid with dimensions M*N, where each cell contains either 0 or 1. Additionally, he is provided with the coordinates of source and destination cells.You can only move to places whose value is 0. Furthermore, he is given the move rule (x, y) which helps in finding the location for the next move. From the given cell, you can move in four directions (forward, back,right ,left), unless they are out of grid. The rules for finding the next move from a current cell are given below.

- For moving forward, add the move rule to the current cell.

- For moving right, from current position add the move rule, rotate the path 90 degree clockwise,

- For moving left, from current position add the move rule, rotate the path 90 degree anticlockwise direction,

- For moving backward, from current position add the move rule, rotate the path 180 degree in clock or anti clockwise.

The rules can be understood better from the following example. Let the current cell be (1,1) and the move rule as (1,2)

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

on forward the next cell would be (2,3) ,
on right the next cell would be (3,0),
left (-1,2) and back (0,-1) are out of grid.

Obeying the given move rule, find out minimum how many cells he need to travel in order to reach the destination.

Constraints

4 <= M, N <= 50

Input

First line consist of two space separated integers M and N denoting the number of rows and columns in the grid.

Next M lines consists of N space separated integers representing the grid.

Next line consists of two space separeted integers denoting source cell.

Next line consists of two space separated integers denoting destination cell.

Last line consists of two space separated integer representing move rule.

Output

Print a single integer denoting the minimum moves required to reach the destination.

Time Limit (secs)

1

Examples

Example 1

Input

6 6

0 1 0 0 0 0

0 0 0 0 0 1

0 1 0 0 0 0

1 1 0 0 0 1

0 0 0 0 0 0

1 1 0 0 1 0

1 0

5 3

1 2

Output

3

Explanation

Akhil needs to move from (1, 0) to (5, 3) and the given step for next move is (1, 2).

In order to minimise the number of moves, he follows the path (1,0) -> (2,2) -> (3,4) ->(5,3) where in total he made 3 moves. No other paths will give moves less than 3. Hence print 3 as the output.

Example 2

input

6 6

0 0 0 0 1 0

0 0 1 0 0 1

0 1 0 1 0 0

1 1 1 0 0 0

1 0 0 0 0 1

1 0 0 1 1 0

0 0

4 4

0 2

Output

4

Explanation

Akhil needs to move from (0, 0) to (4, 4) and the given step for next move is (0, 2).

In order to minimise the number of moves, he follows the path (0,0) -> (0,2) -> (2,2) -> (2,4) -> (4,4) where in total he made 4 moves. No other paths will give moves less than 4. Hence print 4 as th output.

2. XFromYMarks: 30

Problem Description

Malaika is very fond of strings so whenever she gets some free time, she will keep herself engaged in string based activities.

One day, she came across a question where she will be given two strings X & Y and asked to form X from Y. The rules for forming the string are given below.

- The string X should be formed with the concatenation of the sub strings of Y. You can also select the sub strings from Y in reversed order.

- The length of the sub strings selected from Y should be greater than or equal to one.

- Aim is to minimize the number of sub strings that are selected from Y and concatenated to form X.

- A term *String Factor* is defined which is calculated as (number of sub strings selected from Y) * S + (number of sub strings selected from reversed Y) * R, where S and R are given in the input.

- You also have to minimize the *String Factor* while maintaining the minimum number of sub strings.

Given two strings X and Y and two integers S and R, find the minimum *String Factor* of the string X following above rules.

Constraints

1 <= lengths of X,Y <= 10^4

0 <= S, R <= 10^3

X, Y consists of lower case alphabets only.

Input

First line consists of string X.

Second line consists of string Y.

Third line consists of two integers S and R separated by space.

Output

Form the string X from string Y following the above rules and print the *String Factor* of X. Print "Impossible" if X can't be formed from Y.

Time Limit (secs)

1

Examples

Example 1

Input

niveditha

lavekdahnita

3 5

Output

17

Explanation

For forming the string *niveditha* from *lavekdahnita,* select sub strings ni from Y, ve from Y, d from Y, it from Y, ha from reversed Y. No other selections can give less than five sub strings.

String Factor = (number of sub strings selected from Y) * S + (number of sub strings selected from reversed Y) * R = (4*3) + (1*5) = 17

Example 2

Input

abcdef

pafedexycbc

4 2

Output

6

Explanation

For forming the string *abcdef* from *pafedexycbc,* select the sub string 'a' from reversed Y, bc from reversed Y, def from reversed Y. No other selections can give less than three sub strings.

String Factor = (number of sub strings selected from Y) * S + (number of sub strings selected from reversed Y) * R = (0*4) + (3*2) = 6

3. NoMoreSymbolsMarks: 100

Problem Description

Prabhu excels at typing letters but struggles with symbols and numbers. To simplify, he decided to represent mathematical expressions using words. For numbers, he mentions each digit individually with the character 'c' to signify the entire word representing the number. Prabhu exclusively uses lowercase letters as he's not proficient with shift keys or caps lock.

For instance

111 is written as oneconecone

120 is written as onectwoczero

For a single operation: Operation Operand1 Operand2

Example: "add one two" represents 1+2.

For two functions: Operation1 Operation2 Operand1 Operand2 Operand3

Example: "add mul twoctwo threecone two" equals (22*31)+2.

For another variation: Operation1 Operand1 Operation2 Operand2 Operand3

Example: "add oneconecone div onectwoczeroczero twoctwo" equals 111+(1200/22).

Prabhu uses the following operations:

- add for addition (e.g., 2+2=4).

- sub for subtraction (e.g., 2-2=0).

- mul for multiplication (e.g., 2*2=4).

- rem for remainder (e.g., 2%2=0).

- pow for power (e.g., 2^2=4).

To convert and evaluate Prabhu's mathematical expression, output the result in numbers. If any word cannot be resolved as an operation or operand during evaluation, print "expression evaluation

stopped invalid words present" If all words are recognized but the expression cannot be solved, print "expression is not complete or invalid"

Note:

- The input does not contain float or negative numbers.

- Verify the correctness of words first, followed by correctness of expression.

Constraints

0 < Characters in the first line including space < 100

0 < Operands in the expression < 20

Input

Single line denoting the expression.

Output

Single integer repersenting the result of the expression evalutated in numbers not in words.

Time Limit (secs)

1

Examples

Input

add one sub twochundered one

Output

expression evaluation stopped invalid words present

Explanation

In word twochundred, hundred is not a valid word only zero to nine can be used

Example 2

Input

five mul six six fourcninecnine zero

Ouput

expression is not complete or invalid

Explanation

Everywords in the expression is valid but the expression cannot be evaluated only mul operation is found there. After executing, there are still other words are left so the expression is not complete or invalid

Example 3

Input

mul add sub six five oneczero two

Ouput

22

Explanation

The above prabhu expression represents ((6-5)+10)*2

4. ConflictFreeTeamMarks: 100

Problem Description

You are a project manager in a large IT company. You need to select a team of employees to work on a project. You have a list of employees who are eligible for the selection. Employees are indexed from 1 to N.

However, there is a certain rule that must be followed in order to select the team. There are some employees who have some personal conflicts and they can't be in a team together. Also, each employee has a skill value assigned to them, representing their level of expertise. As a project manager, your task is to select a team of employees such that the total expertise of the team should be maximum, keeping the employees incompatibility in mind. Two employees are said to be incompatible if they have any conflicts among them.

Given the employees, their level of expertise value and employee pairs with conflicts, find the maximum possible expertise of the team.

Note: The selected team can contain one employee also.

Constraints

$1 <= n <= 1000$

$0 <= c <= 100$

$1 <= expertise[i] <= 10^4$

Input

First line consists of two integers *n,c* separated by space, representing the number of employees and number of pairs having conflicts.

Next *c* lines, each consists of two integers separated by space. These represents the id of the employees having conflicts between them.

Last line consists of *n* integers separated by space, where i[th] integer represents the expertise level of ith employee.

Output

Print a single integer denoting the maximum possible expertise of the team.

Time Limit (secs)

1

Examples

Example 1

Input

8 6

1 2

2 5

3 6

6 8

1 4

7 8

7 5 4 3 1 6 2 9

Output

21

Explanation

You can form a team with employees [3, 1, 5, 8] who don't have any conflicts among themselves and the total expertise of the team is the sum of skill values of all employees i.e., 4+7+1+9 = 21. No other combination of employees will give an expertise which is greater than 21.

Example 2

Input

10 4

1 5

3 9

2 5

7 10

2 6 3 8 12 9 7 14 1 10

Output

56

Explanation

You can form a team with employees [5, 10, 3, 4, 6, 8] who don't have any conflicts among themselves and the total expertise of the team is the sum of skill values of all employees i.e., 12+10+3+8+9+14 = 56. No other combination of employees will give an expertise which is greater than 56.

5. WaterResourceMarks: 200

Problem Description

Salva is attempting to mitigate water scarcity by installing borewells in regions that have not experienced rainfall for several years. Despite the prolonged drought, the region, consisting of hills and valleys, had ample rainfall in the past. Salva believes that underground water sources still exist in these areas, and by identifying and drilling at these locations, he hopes to mitigate the water shortage.

To determine the optimal borewell locations, Salva has gathered surface data for the region. The goal is to identify the best place for borewells based on the characteristics of the valleys. During rainfall, water accumulates in the valleys and seeps into the ground, making valleys with a wider span more likely to contain underground water.

In the context of this region, a valley is the space between peaks, and a peak is defined as the highest point in the surrounding region - a local maximum in mathematical terms.

Salva's collected surface data is represented by a 2D map given by the equation,

$$Y = \sum_{i=0}^{n-1} \sin\left(A_i x + B_i\right)$$

where Y represents the height of the surface from the measuring level at a distance x from the measuring point. The measuring point is the origin, (0,0), and measurements are taken towards the positive x-axis. The region under consideration has a length of 2pi. For clarity, **consider 2*pi as 6.2831**.

From the provided data, the task is to identify the widest valley. The widest valley is characterized by the maximum horizontal distance between its corresponding peaks or maximas. The output should indicate the count of the valley from the left. If there are multiple valleys with the same width, the leftmost one should be reported. The result should be presented with an accuracy of up to four decimal places.

Constraints

2 <= N <= 25

0 <= Elements of array A and B <= 15

Elements of array A will be unique

Input

First line contains 'n' where n is an integer represent number of element in the arrays.

Second line contains 'n' space separated integers representing elements of array A

Third line contains 'n' space separated integers representing elements of array B

Output

Single integer denoting which valley is suitable to lay the bore counting from the Start. Local maxima should be accurate upto four decimal places.
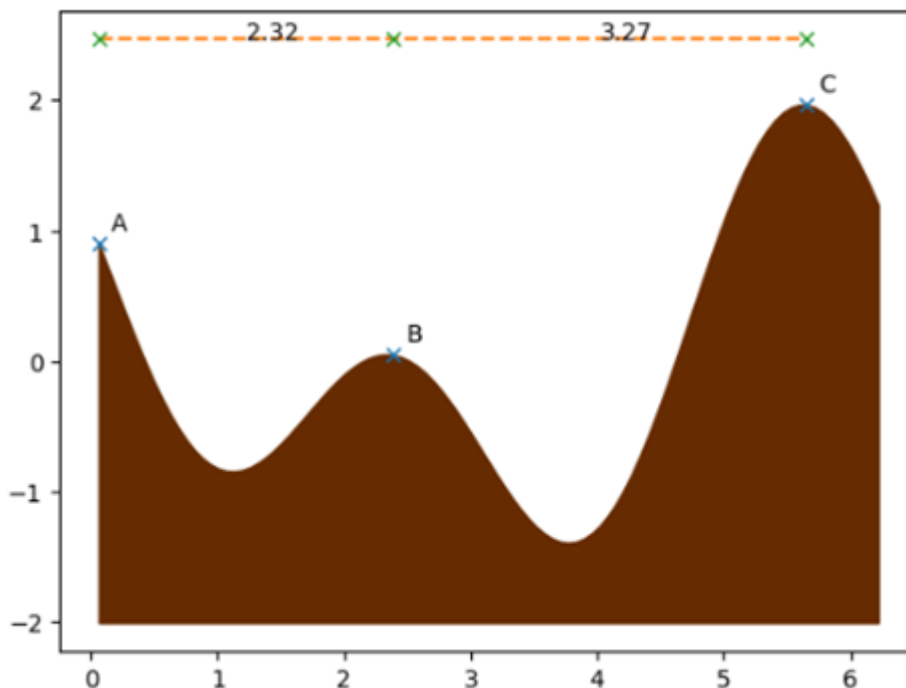
Time Limit (secs)

1

Examples

Input

2

1 2

2 3

Output

2

Explanation

The equation formed by the Arrays are

$y = \sin(x + 2) + \sin(2x + 3)$

The first valley is in between the peak A and B, the second valley is between the peak B and C. The second valley is the widest. So, the output is 2
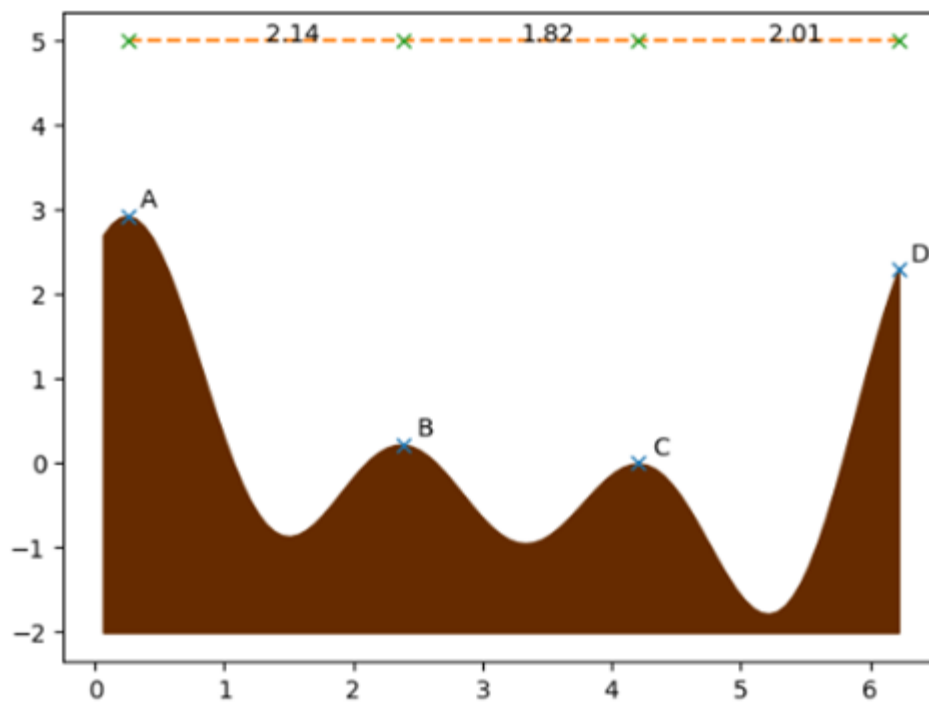
Example 2

input

3

1 2 3

1 1 1

output

1

Explanation

The equation formed by lines are

$$y = \sin(x + 1) + \sin(2x + 1) + \sin(3x + 1)$$

It represents the surface below. And thus the output 1.

The first valley is in between the peak A and B, the second valley is between the peak B and C, the third valley is between C and D. The first valley is the widest.

6. 9X9

Problem Description

We have a classic number-placement puzzle for you! This puzzle is played on a 9x9 matrix, divided into nine 3x3 sub matrices.

In this matrix some cells are pre-filled, some cells are provided with hints, while the other cells are empty. You have to fill these empty cells. Rules for filling the puzzle are given below.

- Every row should have 1-9 numbers without repetition.

- Every column should have 1-9 numbers without repetition.

- Every 3*3 sub matrix should have 1-9 numbers without repetition. The sub matrices are highlighted with thick boarders in the given figures in example section.

- There are some cells which are provided with hints i.e., those cells should be filled with one of the numbers from the given list in the input.

Tina is participating in this puzzle competition, and due to time constraints, in the last minutes, she quickly filled in the puzzle but might have made some mistakes. But the organizers felt none of them might have solved correctly and decided to rank the participants based on the number of cells that require replacements with other numbers to make the grid correct. The fewer the count of such cells, the better their rank will be. Fortunately, if Tina filled the whole thing correctly, then she will win the puzzle. If the count of cells that need modifications is <= given K, then Tina will get a rank and has the chance of winning. Judge will compare the closest solution based on individual answer to minimize the number of cells need to be changed. Its impossible for her to win, if count of cells that need modifications is greater than given K, because she won't get into the ranking list.

Given puzzle filled by Tina, a list of numbers, and an integer 'K', determine whether Tina has a chance of winning the competition or not.

Note : You can only change the hinted and empty cell values but not pre-filled cells in order to make the grid correct.

Constraints

Grid size remains same i.e., 9x9.

Grid values will be basically from 1-9. But we use leading 0s to represent empty cells and trailing 0s to represent hinted cells.

1 <= K <= 10

1 <= len(list) <= 5

Input

First 9 lines consists of 9 space separated numbers denoting the matrix, filled by Tina where,

- If a number has a leading zero, it means the cell was filled by Tina.

- If a number has a trailing zero, it indicates that it's a hinted cell, and it should be filled with one of the numbers from the given input list.

- If no zero at all, its a pre-filled cell whose value can't be changed.

10th line consists of a list of numbers which are allowed to be placed into the hinted cells.

The 11th line of input contains an integer 'K,' which represents the maximum number of cells that are allowed to have incorrect values while still being eligible for a ranking in the competition.

Output

- Print "Won" if Tina placed everything correctly else,

- Print "Impossible" if it's impossible for Tina to win the competition based on the given criteria else,

- If she didn't win but has a chance of winning, print the indices of the cells that are wrongly placed, arranged from top to bottom and left to right, each on a single line (0 based indexing).

Time Limit (secs)

1

Examples

Example1

Input

06 1 09 7 4 02 03 5 08

4 50 7 08 03 01 6 09 2

8 2 04 6 09 05 01 7 4

20 03 06 04 1 09 5 08 07

5 09 01 2 07 08 04 06 3

07 08 4 03 5 06 20 01 9

9 6 02 01 08 3 07 4 5

3 04 05 09 06 07 8 20 1

01 7 08 50 2 4 01 3 6

5 7 2

3

Output

2 2
8 6

Explanation

Given grid is visualized below with the following color codes.

Green - Pre-filled cells

White - Filled by Tina

Orange - Hinted cells

The above grid which is filled by Tina is incorrect because of the following reasons.

- 4 is present twice in the first 3*3 sub grid, 3rd column and also in 3rd row (1-based indexing, from top left)

- 1 is present twice in the last 3*3 sub grid (9th), 7th column and also in 9th row (1-based indexing, from top left)

In order to make this grid correct in minimum replacements, we will place 3 in the position (2,2) and 9 in the position (8,6). Since the number of cells that are needed to be modified is less than K, she holds the chance of winning and hence we print the cells indices which require modifications, in left to right, top to bottom order.

Example 2

Input

5 3 04 60 7 08 09 01 01

6 07 02 1 9 5 03 04 08

01 9 8 03 04 02 05 6 07

8 05 09 70 6 01 04 02 3

4 02 60 8 05 3 07 09 1

7 01 03 09 2 04 08 05 6

09 6 10 05 01 07 2 8 04

02 08 07 4 1 9 06 03 5

07 04 05 02 8 06 10 7 9

6 7 1 2

2

Output

Impossible

Explanation

Given grid is visualized below with the following color codes.

Green - Pre-filled cells

White - Filled by Tina

Orange - Hinted cells

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

1 is present twice in the 1st row, 3rd 3*3 submatrix and 9th column. Again 1 is repeated in the 7th row, 8th 3*3 submatrix and 5th column. Also 7 is repeated in the 1st column,9th row as well as in the 7th 3*3 sub grid (1-based indexing). Thus it can't be corrected using 2 replacements. Hence print "Impossible".