# Getaway Gala

## Problem Description

At the annual office holiday party, excitement buzzed as employees gathered for a lucky draw. Among the prizes was a weekend getaway voucher.

With a small number of attendees forming a row, the organizer chose to forego the conventional lucky dip approach. Instead, they introduced a more curious and engaging method that will eliminate every employee except one, who is considered as winner.

The organizer aimed to eliminate employees in two rounds. In the first round, he will form a row of employees who participated. Then he will select the first letters of all the employees from left to right and form a string S. Then he will start eliminating the employees whose name is forming the palindrome (every time he will delete the first name from left to right which is forming a palindrome). He will continue this process till the string S does not have any sub strings which are palindromes.

For example, consider the names {Hari Giri Siri Gopi Hima} in the row. Here the string S formed by picking the first letter of the names is HGSGH. Note that substring GSG is a palindrome .When processing from left to right, the alphabet G from the name Gopi forms a palindrome. We call the name *Gopi* as *Mirror Word*. Hence we eliminate Gopi in the first iteration.

Now only names {Hari Giri Siri Hima} remain. String S is HGSH which does not have any palindrome. Hence the final row will be {Hari Giri Siri Hima}. If string S had more palindromes we would apply the same procedure as mentioned in paragraph above. Since string S is now palindrome free, the organizers will now apply a different criteria.

This criteria will be to remove every $N^{th}$ person from the remaining names everytime processing the names from left to right. The last name remaining is the winner of the lucky draw.

Given the list of names of employees, and the value of N, find out the who the winner is.

## Constraints

Names comprise of upper and lower case characters. Processing is case insensitive.

1 <= number of employees <= 1000

1 <= N <= 1000

1 <= length of name <= 10

## — Input

First line consists of an array indicating the names of employees present in the row. Names are space separated.

Second line consists of a single integer N denoting the interval of elimination.

## — Output

Print the name of the employee who is going to be the winner. Print the name as it is given in the array.

## — Time Limit (secs)

1

## — Examples

Example 1

Input

Janu gita sana gopi jaslin Tony Ritu Naina sonu Neha

2

Output

Janu

Explanation

As we can see, gopi is forming a mirror word according to the given rules, thus we eliminate the employee named gopi. Now gopi's position will be empty. Again, the name Neha is forming mirror word, hence we remove the name Neha and the resulting row will be {Janu gita sana jaslin Tony Ritu Naina sonu}. Now there can be palindromes after picking the first alphabets of the remaining names. Hence the second criteria is now applied where N is 2. Now, processing from left to right, if we st

codevita.tcsapps.com/main_page.jsp

Submissions

Submissions

missions

# Traffic Flow

## Problem Description

A traffic flow grid is a conceptual representation used in transportation planning and traffic engineering to model and analyze the movement of vehicles through a network of intersections in the city called as junctions. A traffic flow grid is akin to a matrix, with rows and columns denoting various junctions, and matrix cell values symbolizing the traffic flow in that junction. In the era of screens and swipes, traffic's stage can be seen on mobile!

Srihan finds himself caught in traffic on his way to the workplace. To speed up his journey, he opts to assess the traffic flow grid and chooses a quicker route. He currently resides at coordinates (i, j), and his office is located at coordinates (a, b) in the traffic grid. Given a rule that from a given junction, he can only go to right or down junction (junction is a cell, in terms of matrix). Could you assist him in finding a faster route to reach his workplace?

## Constraints

Note that the provided positions in the input are indexed starting from 1, where left top is (1, 1).

1 <= n, m <= 100

0 <= traffic flow <= 10^3

## Input

First line consists of two space separated integers n, m denoting the number of rows and columns in the traffic grid.

Next n lines consists of m space separated integers denoting the traffic flow. The value at i, j indicates the traffic flow at junction i and j.

Line n+2 consists of two space separated integers denoting the coordinates of Srihan's current position.

Line n+3 consists of two separated integers denoting the coordinates of Srihan's workplace.

## Output

Single integer denoting the the minimum traffic flow Srihan must navigate to reach his workplace. In case there is no path from his current location to workplace, pr
"-1".

Search

ENG
IN

12:24
20-01-2024

codevita.tcsapps.com/main_page.jsp

— Time Limit (secs)

1

— Examples

Example 1

Input

4 4

3 1 9 2

12 3 10 1

11 10 14 2

5 4 3 7

1 2

4 4

Output

22

Explanation

Srihan will choose the path - (1,2) -> (1,3) -> (1,4) -> (2,4) -> (3,4) -> (4,4)

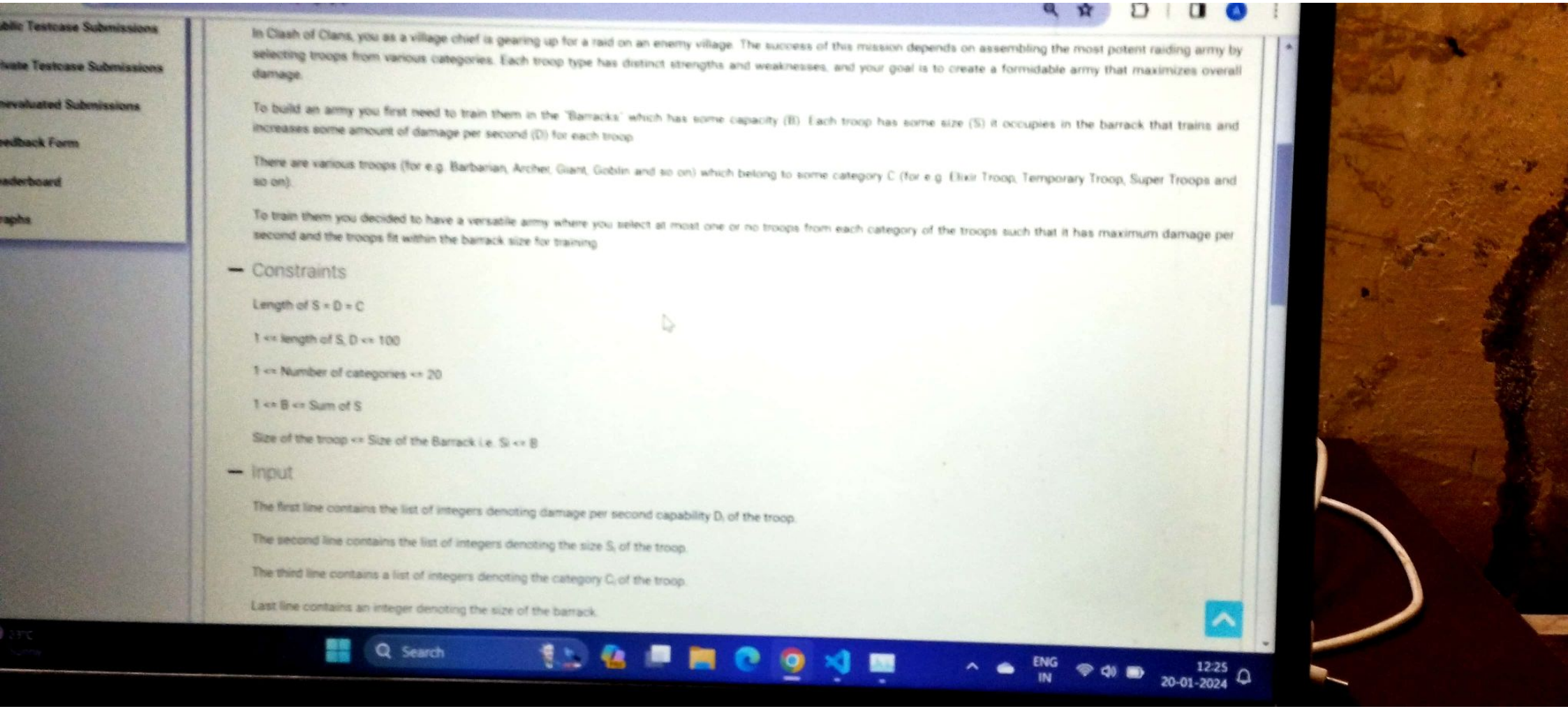
The total traffic flow Srihan navigated is 1 + 9 + 2 + 1 + 2 + 7 = 22 which is the minimum possible.

Example 2

Search

ENG

12:24


ublic Testcase Submissions
ivate Testcase Submissions
nevaluated Submissions
edback Form
aderboard
aphs
In Clash of Clans, you as a village chief is gearing up for a raid on an enemy village. The success of this mission depends on assembling the most potent raiding army by selecting troops from various categories. Each troop type has distinct strengths and weaknesses, and your goal is to create a formidable army that maximizes overall damage.
To build an army you first need to train them in the 'Barracks' which has some capacity (B). Each troop has some size (S) it occupies in the barrack that trains and increases some amount of damage per second (D) for each troop.
There are various troops (for e.g. Barbarian, Archer, Giant, Goblin and so on) which belong to some category C (for e.g. Elixir Troop, Temporary Troop, Super Troops and so on).
To train them you decided to have a versatile army where you select at most one or no troops from each category of the troops such that it has maximum damage per second and the troops fit within the barrack size for training.
Constraints
Length of S = D = C
1 <= length of S, D <= 100
1 <= Number of categories <= 20
1 <= B <= Sum of S
Size of the troop <= Size of the Barrack i.e. Si <= B
Input
The first line contains the list of integers denoting damage per second capability D, of the troop.
The second line contains the list of integers denoting the size S, of the troop.
The third line contains a list of integers denoting the category C, of the troop.
Last line contains an integer denoting the size of the barrack.
Search
ENG
IN
12:25
20-01-2024

## — Output

Print the maximum damage per second that can be achieved.

## — Time Limit (secs)

1

## — Examples

Example 1

Input

8 9 4 9 1 8 1 5 6 8

2 5 7 2 3 4 5 9 3 8

4 2 2 3 4 3 2 1 2 1

10

Output

26

Explanation

The goal is to maximize the damage per second where you select at most one or none from each category. So here we choose the $1^{st}$ troop which belongs to category 4, $2^{nd}$ troop which belongs to category 2 and $4^{th}$ troop which belongs to category 3 whose total size is 9 which is within the barrack size. We could not accommodate any troop from category 1 because the damage per second capability reduces or the barrack capacity falls short. Hence, the total damage per second is 26.

Example 2

Input

# Rat In A Labyrinth

## Problem Description

Jenny is in the process of creating a unique cardboard labyrinth composed of various room-like structures, all made from cardboard pieces. Her objective is to ensure that the entrances to these rooms are exclusively positioned along the outer edges of the labyrinth.

To begin, Jenny starts with a rectangular piece of cardboard with dimensions "m" units in length and "n" units in height. She also has several straight cardboard pieces at her disposal. Jenny's plan is to insert these cardboard pieces inside the rectangular cardboard in a manner that each inserted piece runs parallel to one of the rectangle's sides. This arrangement will form distinct room-like structures within the labyrinth. Jenny prefers rooms that have precisely four sides, and she designs her labyrinth with such rooms exclusively. Once the labyrinth is complete, Jenny is eager to play with it and decides to place her pet rat inside for a playful adventure. Now, she aims for finding the percentage probability of the rat successfully exiting the labyrinth rounded to the nearest integer value.

Given that the lower-left corner of the labyrinth is situated at coordinates (0,0) within quadrant one, and with knowledge of the labyrinth's length, height, the placement of straight cardboard pieces within it, print the percentage probability of the rat successfully exiting the labyrinth rounded to the nearest integer value.

**Note** : It is assured that the lines are valid and unique.

## Constraints

$0 <= x1, x2, y1, y2 <= 100$

$0 <= m,n <= 100$

## Input

First line consists of two space separated integers, m and n denoting the length and height of the cardboard box.

Second line consists of an integer T, denoting the number of straight card board pieces that are inserted in the cardboard box as stated above.

Next T lines will have 4 integers denoting the starting and ending points of the straight cardboard pieces where they are placed. First two integers denotes (x1,y1) and the next two denotes (x2,y2).

## Output

Print the percentage probability of the rat successfully exiting the labyrinth rounded to the nearest integer value.

## Time Limit (secs)

1

## Examples

Example 1

Input

4 4

5

1 0 1 3

0 3 4 3

1 2 2 2

2 0 2 3

1 1 2 1

Output

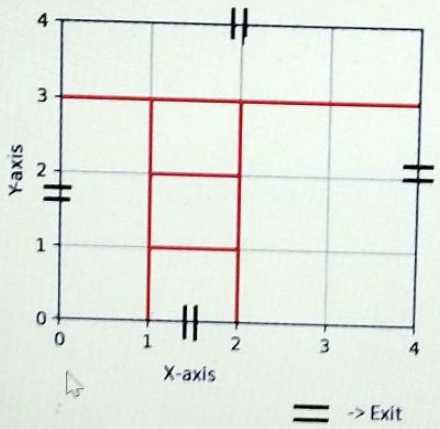
67

Explanation

If we visualize the the above labyrinth, it will be like below.

codevita.tcsapps.com/main_page.jsp

Top view of the cardboard box




The given lines divides the cardboard into 6 parts among which 2 parts have no entrance. Thus escape rate of rat from the labyrinth will be number of rooms from which rat can exit)/ (total number of rooms)*100 i.e., (4/6) * 100 which is (0.66666..) * 100 = 66.666.., which on rounding off to the nearest integer will be 67.

Example 2

Input

3 4

3

1 0 1 4

2 1 2 4

1014

2124

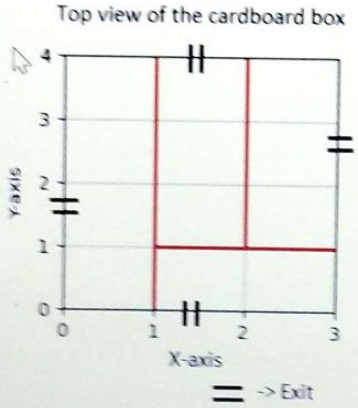
1131

Output

100

Explanation

If we visualize the the above labyrinth, it will be like below.




The given lines divides the cardboard into 4 parts and all parts have entrance. Thus escape rate of rat from the labyrinth will be number of rooms from which rat c
exit)/ (total number of rooms)*100 i.e., (4/4) * 100 which is (1) * 100 =100, which on rounding off to the nearest integer will be 100.

codevita.tcsapps.com/main_page.jsp

Submissions

Submissions

missions

# Room Optimization

## — Problem Description

Vishal, a builder focused on constructing rooms along the port side, acquires land with the goal of optimizing the number of rooms he can construct with that space. The rooms are limited to four specific types, categorized as k-square unit rooms, where k can take on one of the values {A, B, C, D}.

At first, Vishal had the opportunity to purchase X units of land. Subsequently, he had the chance to acquire an additional Y units of land, which is contiguous to the initial plot.

He follows two rules while constructing which are given below.

- He aims for constructing more number of rooms within that area, irrespective of the size of the rooms.
- If there are multiple combinations of rooms possible within given area and maximum number of rooms is same for all combinations, then he will opt for the combination with larger room area. For example if the area is 21 units, and the values of {A, B, C, D} are {5, 9, 7, 11} then we have (7, 7, 7), (9, 7, 5), (11, 5, 5). Here the combination (11, 5, 5) is comprises of three rooms with largest room size be 11.since Vishal prefers larger rooms given the total number of rooms is same, this combination will be constructed.

Given the values of {A, B, C, D}, the initial land Vishal bought and the additional units of land he acquired, calculate the difference in the number of rooms of each type resulting from the additional Y units of land.

## — Constraints

1 <= A, B, C, D <= 1000

A, B, C, D will be Unique.

1 <= (X+Y) <= 1000

## — Input

First line consists of four space separated integers denoting the values of A, B, C and D.

First line consists of four space separated integers denoting the values of A, B, C and D.

Second line consists of two space separated integers denoting the values of X and Y.

## Output

Four space separated integers denoting the difference in the number of rooms of each type in the same order given in the input. Note that the difference will be calculated as (number of rooms of Type-N before extension) - (number of rooms of Type-N after extension).

## Time Limit (secs)

1

## Examples

Example 1

Input

9 11 7 5

21 6

Output

0 1 -1 -2

Explanation

Given initial area is 21 units. Vishal can plan to build 1 room of 11 square units and 2 rooms of 5 square units which will occupy 1 * 11 + 2 * 5 = 21 square units. But the extended area is 6 units, which means the total area in which he can build rooms is 21 + 6 = 27 square units. For maximizing the number of rooms within 27 square units, he can build 1 room of 7 square units and 4 rooms of 5 square units which will occupy 1 * 7 + 4 * 5 = 27 square units.

Hence the final composition of rooms before and after extension is -

codevita.tcsapps.com/main_page.jsp

| Room Type | Number of rooms selected (Before Extension) | Number of rooms selected (After Extension) |
|---|---|---|
| 9 square units | 0 | 0 |
| 11 square units | 1 | 0 |
| 7 square units | 0 | 1 |
| 5 square units | 2 | 4 |

Hence the difference in the number of rooms of each type is 0 1 -1 -2.

Example 2

Input

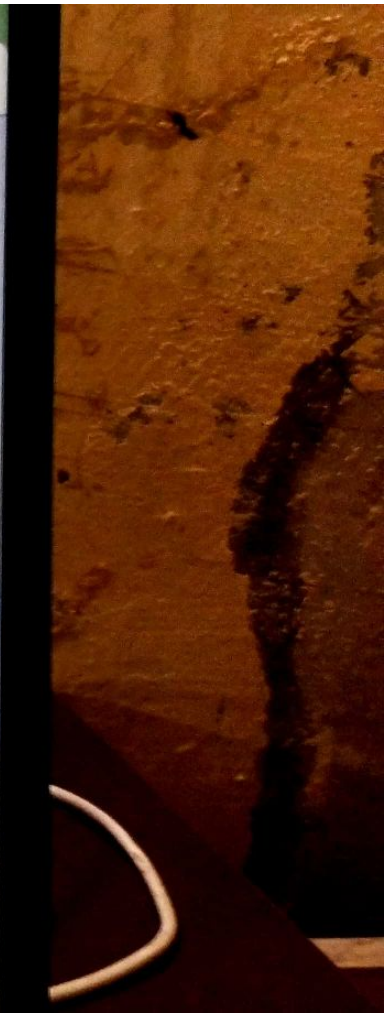
2 5 13 6

14 9

Output

-2 -1 0 0

Explanation



Explanation

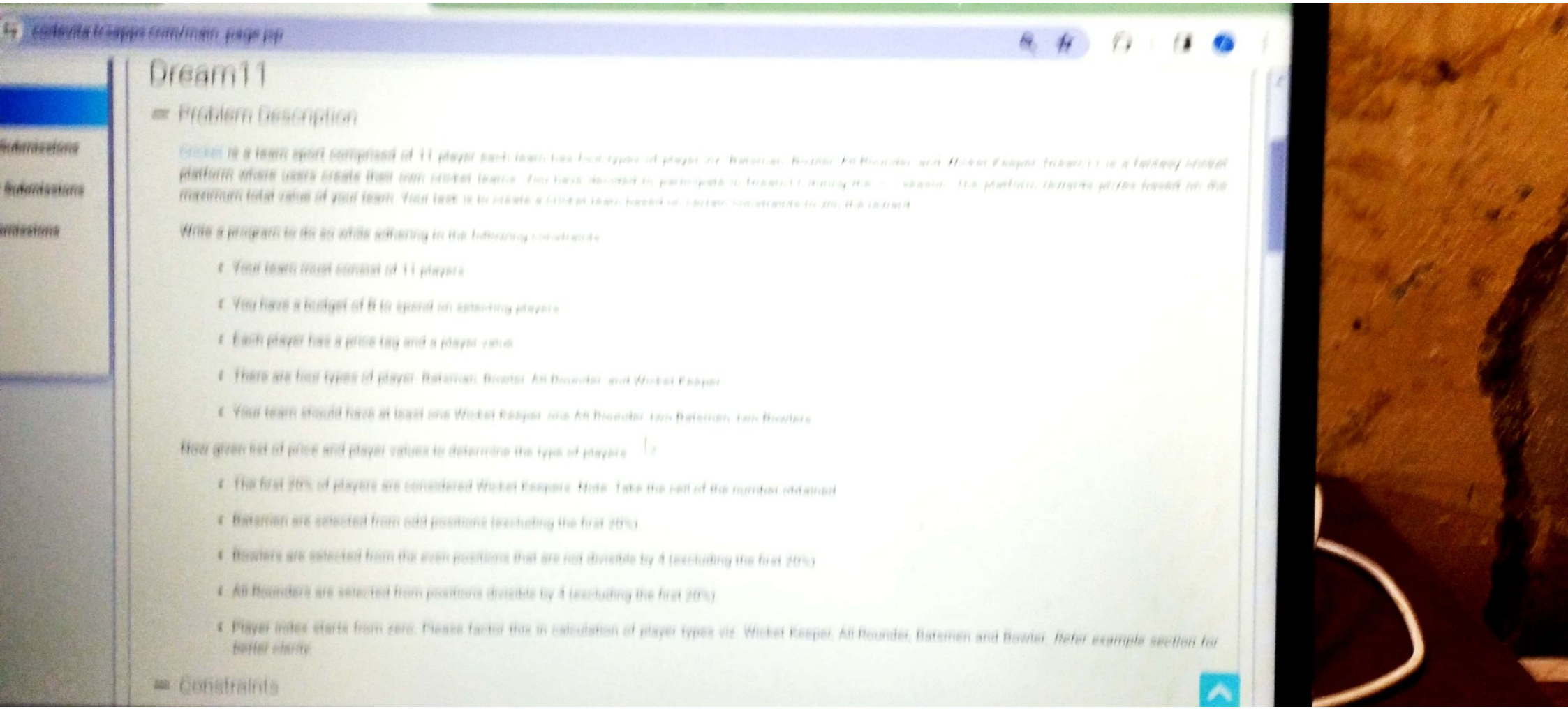
Given initial area is 14 units. Vishal can plan to build 7 rooms of 2 square units which will occupy 7 * 2 = 14 square units. But the extended area is 9 square units, which means the total area in which he can build rooms is 14 + 9 = 23 square units. For maximizing the number of rooms within 23 square units, he can build 9 rooms of 2 square units and 1 room of 5 square units which will occupy 9 * 2 + 1 * 5 = 23 square units.

Hence the final composition of rooms before and after extension is -

| Room Type | Number of rooms selected (Before Extension) | Number of rooms selected (After Extension) |
|---|---|---|
| 2 square units | 7 | 9 |
| 5 square units | 0 | 1 |
| 13 square units | 0 | 0 |
| 6 square units | 0 | 0 |

Hence the difference in the number of rooms of each type is -2 -1 0 0

Upload Solution [ Question : F ]


Dream11
Problem Description
Constraints

## Constraints

$11 <= N <= 200$

$1 <= B <= 1000$

$1 <= P <= 20$

$1 <= V <= 20$

## Input

The first line contains N the total number of players.

The second line consists of vector P which denotes the list of player prices, where the $i^{th}$ element represents the price of $i^{th}$ player.

Third line consists of vector V which denotes the list of player value, where the $i^{th}$ element represents the values of $i^{th}$ player.

The last line contains B the total budget.

## Output

Print the maximum total value M which is the summation of the selected player values, that can be obtained while satisfying all the constraints and is within the budget else print "Insufficient Budget".

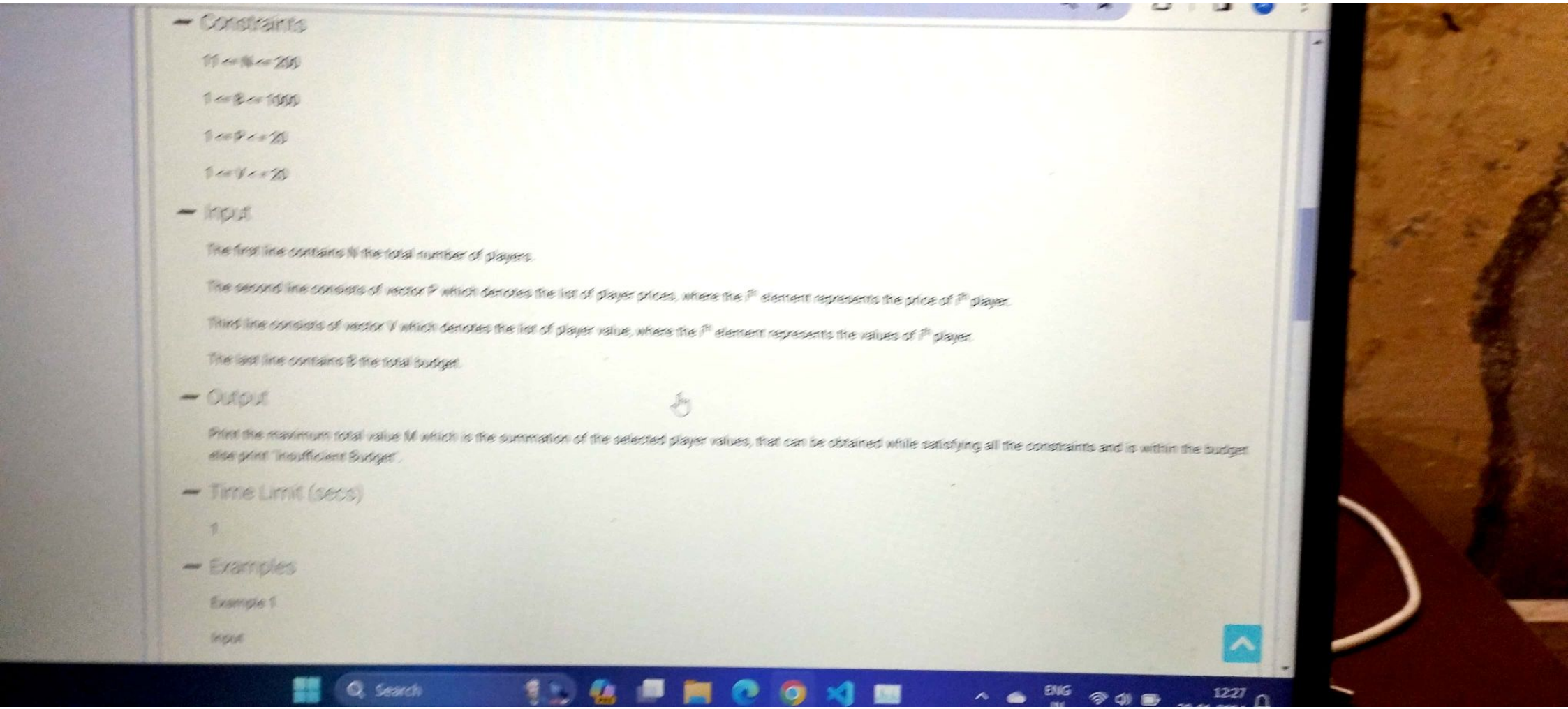
## Time Limit (secs)

1

## Examples

Example 1

Input



codevita.tcsapps.com/main_page.jsp

One needs to have a total of 11 players in the team, so we need to exclude 1 player and form the team. So, the player that is excluded is the player at index 7 with price 8 and value 6. So, the total max value that can be obtained is 34 with the total price of 50 which is within the budget while maintaining the constraint on player type viz. at least 2 batsmen and 2 bowlers and 1 wicket-keeper and 1 all-rounder.

Refer the below diagram for selected players.

| Index | Price | Player Value | Player Type |
|---|---|---|---|
| 0 | 4 | 2 | Wicket Keeper |
| 1 | 3 | 1 | Wicket Keeper |
| 2 | 3 | 1 | Wicket Keeper |
| 3 | 6 | 2 | Batsman |
| 4 | 5 | 3 | All Rounder |
| 5 | 2 | 6 | Batsman |
| 6 | 8 | 1 | Bowler |
| 8 | 2 | 2 | All Rounder |
| 9 | 7 | 7 | Batsman |
| 10 | 8 | 6 | Bowler |
| 11 | 2 | 3 | Batsman |
| Total | 50 | 34 | |

Example 2

Input

12

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

2 1 1 2 3 6 1 6 2 7 6 3

50

Output

Insufficient Budget

12

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

2 1 1 2 3 6 1 6 2 7 6 3

50

Output

Insufficient Budget

Explanation

Consider the following diagram:

| Player | Price | Player Value | Player Type |
|---|---|---|---|
| 0 | 4 | 2 | Wicket Keeper |
| 1 | 3 | 1 | Wicket Keeper |
| 2 | 3 | 1 | Wicket Keeper |
| 3 | 6 | 2 | Batsman |
| 4 | 5 | 3 | All Rounder |
| 5 | 2 | 6 | Batsman |
| 6 | 8 | 1 | Bowler |
| 7 | 8 | 6 | Batsman |
| 8 | 2 | 2 | All Rounder |
| 9 | 7 | 7 | Batsman |
| 10 | 9 | 6 | Bowler |
| 11 | 2 | 3 | Batsman |

As can be seen the player type has been allocated according to the index.

One needs to have a total of 11 players in the team where one needs to have at least 2 batsmen, 2 bowlers, 1 wicket-keeper and 1 all-rounder. To form such a te requires a minimum budget of 51 which violates the constraints on the budget. Hence, print "Insufficient Budget" as the output.

codevita.tcsapps.com/main_page.jsp

# Chandrayaan - 3

## Problem Description

Recently, India embarked on an ambitious space mission known as Chandrayaan - 3, following its predecessors Chandrayaan - 2. Chandrayaan - 3's mission objective is to explore the Moon's uncharted south pole region, an area shrouded in mystery and scientific intrigue. To achieve this, India deployed a highly advanced lunar rover, aptly named "Pragyan."

Pragyan's software is very sophisticated. It will first simulate the path that it wants to take. However it knows that this approach may result in loopy paths which is detrimental to the distance it can travel. So then it will optimize the path to remove any loops. The method applied to remove that loop is that, at the point of intersection, the rover's direction would change towards the direction the intersected path was heading, and the remaining distance after the intersection would override the intersected path distance. The path would only intersect from the adjacent direction, not the opposite.

Then on this optimized path it tries to calculate the number of craters that it will have to cross subject to a constraint on the distance it can travel from the origin. Origin is the starting coordinate from where the rover would begin its journey.

If at any point, the rover's distance from the origin exceeds the threshold distance (k), it will ignore that travel instruction and continue with the next instruction.

Once the path was finalized, the rover would actually follow it, encountering and crossing craters along the way. The craters are represented in the form of squares, with the possibility of multiple craters overlapping each other but not completely. A crater is considered crossed if the rover crossed it on any of the edges, touched any point on its edge or drove over the edge(s).

## Constraints

Threshold Distance >=1

1 <= Number of Instructions <= 1001

1 <= Number of Craters <= 1000

Size of the Side of the crater >=2

## Input

Search

First line contains the starting coordinate of the rover i.e. Origin.

Second line contains the threshold distance K of the rover from the Origin.

Third line contains N the number of instructions.

Next N line consists of movement instructions for the rover, where each instruction consists of a direction N, E, S, W followed by a positive integer representing the distance to be traveled in that direction.

(N+1)$^{th}$ line contains the numbers of craters C.

Next C lines contain craters in the form X Y S, where X,Y = x and y coordinate of the center of crater and S the size of the sides of the crater.

## Output

Print the number of craters crossed.

## Time Limit (secs)

1

## Examples

Example1

input

2 1

10

3

N2

10

3

N2

E3

N4

2

4 2 2

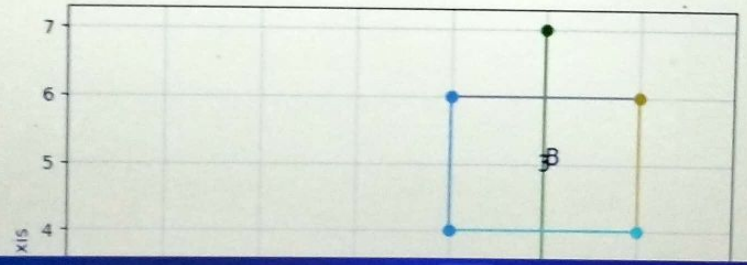
5 5 2

Output

2

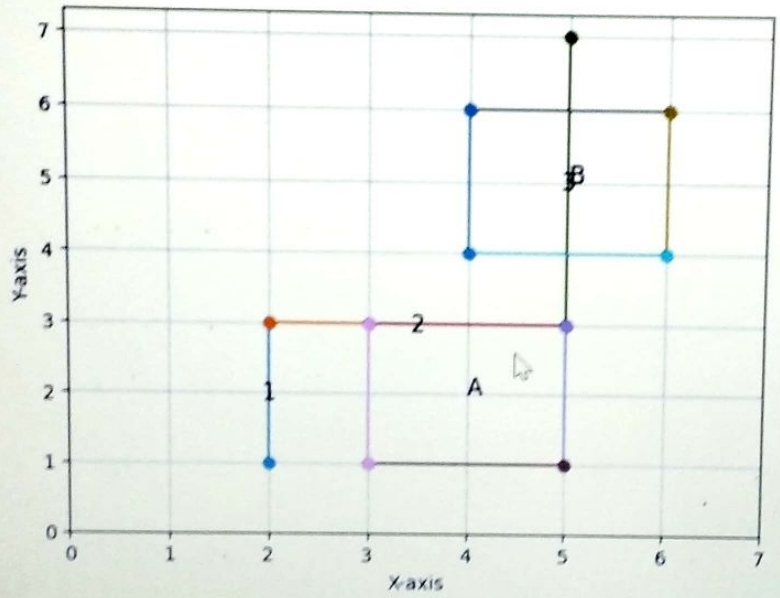
Explanation

The origin from where the rover will start its mission is (2,1 ) and it should not travel beyond a distance of 10 units from the origin. There are 3 instructions that formed the path of the rover which needs to be optimized. There are two craters of size 2 at (4, 2) and (5, 5).



Explanation

The origin from where the rover will start its mission is (2,1 ) and it should not travel beyond a distance of 10 units from the origin. There are 3 instructions that formed the path of the rover which needs to be optimized. There are two craters of size 2 at (4, 2) and (5, 5).




Consider the above diagram where the line segment 1, 2, 3 are the paths of the rover, and the square box A and B are the craters. Since in path 1, 2 and 3 there are no loops / intersections the path is already optimized. Also since all points in the path are within 10 units from the origin this path can be traversed.

When the rover actually travels on this path it crosses both the craters as seen from above diagram. Hence the output is 2.