Help MLA 2

Problem Description

Imagine you are an MLA of a district and there are N number of villages in your constituency.

Your job is to vaccinate all the people in your constituency in minimum amount of time. There are two centres where vaccination is going on. First centre takes m1 minutes as average time for vaccinating one person and second centre takes m2 minutes as average time.

Population of every village is known to you prior to the vaccination drive. Schedule all the villagers to any centre such that overall time for vaccinating all the people of all the villages will be minimum.

Assume that there is no wait time in between vaccinating two people. Also, people belonging to the same village will need to be vaccinated in the same centre.

For example:

First centre takes 2 min as average time

Second centre takes 4 min as average time

Population data of 3 villages is known: 10 30 20

Number of people in 3 villages is given:

v1 = 10, v2 = 30, v3 = 20

Consider if schedule is drawn by distributing equal number of people to both centres, then

First centre: 10 20 total time = (10 + 20) \* 2 = 60 min

Second centre: 30 total time = (30) \* 4 = 120 min

Hence, minimum time required to vaccinate all the people will be = 120 min. i.e., Maximum of time taken in first centre or second centre.

But if it is scheduled like this:

First centre: 10 30 total time = (10 + 30) \* 2 = 80 min

Second centre: 20 total time = (20) \* 4 = 80 min

Minimum time required to vaccinate all the people will be = 80 min

Your job is to schedule these villages such that vaccination time should be minimum.

Constraints

0 < m1, m2 <= 20

0 < N < 100

0 < Population of village <= 100

Input

First line contains an integer m1 which is average time in minutes taken for vaccination by the first centre

Second line contains an integer m2 which is average time in minutes taken for vaccination by the second centre

Third line contains an integer N which is number of villages in the constituency

Fourth line contains N space delimited integers denoting the population of villages

Output

Print the villages which are scheduled at centre1 on first line and the villages which are scheduled at centre2 on second line. For better understanding refer *Examples* sections.

NOTE: - There are multiple answers possible for a given input. As long as your output meets all the conditions, any answer is acceptable.

Time Limit (secs)

1

Examples

Input

2

3

5

10 50 20 30 40

Output

10 50 30

40 20

Explanation:

Given the data of room1 and room2:

First room takes 2 min as average time. Second room takes 3 min as average time. Number of villages in your constituency are 5.

Number of people in each of the 5 villages is given: 50 10 20 30 40

v1 = 50, v2 = 10, v3 = 20, v4 = 30, v5 = 40

If the schedule looks like this:

First room: 10 50 total time = (10 + 50) \* 2 = 120 min

Second room: 30 40 20 total time = (20 + 40 + 20) \* 3 = 240 min

Minimum time required to vaccinate all the people will be = 240 min

But if the schedule is drawn like this:

First room: 10 50 30 total time = (10 + 50 + 30) \* 2 = 180 min

Second room: 40 20 total time = (40 + 20) \* 3 = 180 min

Minimum time required to vaccinate all the people will be = 180 min

And output will be

10 50 30

40 20

Other possible outputs are:

|  |  |  |
| --- | --- | --- |
|  | First Line of Output | Second line of Output |
| Solution 1 | 30 10 50 | 20 40 |
| Solution 2 | 10 50 30 | 40 20 |

There could possibly be more solutions.

In all these cases time required to vaccinate the villagers is same and is the minimum.

Example 2

Input

1

2

3

100 90 70

Output

100 70

90

Explanation:

Given the data of centre1 and centre2:

First room takes 1 min as average time. Second room takes 2 min as average time. There are 3 villages in your constituency.

Number of people in each of the 3 villages is given: 100 90 70

v1 = 100, v2 = 90, v3 = 70

If the schedule looks like this:

First room: 100 90 total time = (100 + 90) \* 1 = 190 min

Second room: 70 total time = (70) \* 2 = 140 min

Minimum time required to vaccinate all the people will be = 190 min

But if the schedule can be drawn like this:

First room: 100 70 total time = (100 + 70) \* 1 = 170 min

Second room: 90 total time = (90) \* 2 = 180 min

Minimum time required to vaccinate all the people will be = 180 min

And the output is:

100 70

90

Other possible output is

70 100

90

In both cases time required to vaccinate the villagers is minimum.

Faulty Keyboard

Problem Description

Mr. Wick has a faulty keyboard. Some of the keys of the keyboard don't work. So, he has copied all those characters corresponding to the faulty keys on a clipboard from some existing document. Whenever those characters need to be typed, he pastes it from the clipboard. In typing whatever is required he needs to make use of paste, backspace and cursor traversal operations. Help him to minimize the number of operations he needs to do to complete his typing assignment. Each operation has one unit weightage.

Mr. Wick prefers verbal clarity over optimization of labour. That's why he prefers to fully process one word before processing the next word. This preference of his is very important to be honoured. Please see Example 1 in the examples section for better understanding.

Constraints

1 <= T <= 10^4

1 <= S <= 16

String T and S will only be comprised of letters a-z and digits 0-9

Input

First line contains text T to be typed

Second line contains string S of all the faulty keys pasted on clipboard

Output

Print the minimum number of operations required for typing the text T

Time Limit

1

Examples

Example 1

Input

experience was ultimate

ew

Output

14

Explanation

experience =(2+2+2+2) =[ {p+b} + {p+b} +{p+b} +{p+b} ]

was=(4)=[ p+m+b+m]

ultimate=(2)=[ p+b ]

where p=paste, b=backspace, m= move cursor

This is the right answer according to Mr. Wick's preference. However, this can also be done in another way. Since, this method does not honour Mr. Wick's preference this will be the wrong answer.

experiencew  =(2+2+2+1) =[ {p+b} + {p+b} +{p+b} +{p}]

as =(0) =[]

ultimate=(2)=[ p+b ]

By this method the number of operations is only 9. However, this violates the constraint of fully processing one word before processing the next word. Hence, this is the wrong answer.

Example 2

Input

supreme court is the highest judicial court

su

Output

17

Explanation

supreme =(1) =[ p]

court=(4)=[ p+m+b+m]

is=(2)=[ p+b ]

the=(0)

highest=(2)= [p+b]

judicial=(4)= [p+m+b+m]

court=(4)= [p+m+b+m]

Train Track

Problem Description

Station master of Codington railway station is working on automation of trains to platform allocation. This allocation will not only prevent manual errors but also bring about a consistency in train to platform allocation. This will improve the passenger experience. Your job is to help the station master achieve the objectives.

The data and the rules that need to be considered for this automation are as follows-

* Codington is a large railway junction. Hence assume that there is always a platform at which the incoming train can be accommodated (infinitely many platforms). However, operating using a less number of platforms is economical and hence preferred.
* Safety is of paramount importance. Hence, safety cannot be traded off for cost. For example, if Train A's departure time is x and Train B's arrival time is x, then we can't accommodate Train B on the same platform as Train A.
* The trains to platforms allocation rules are as follows-
  + The platform which is being freed earlier has to be considered earlier for allocation to the next train. For example, if platform #2 is freed at t=5 and platform #1 is freed at t=6, in that case the next train arriving at t=7 will arrive at platform #2.
  + If two or more platforms are freed at the same time, then the train arrives on the platform with the lowest number. For example, if platform #1 and #2 are freed at t=5 and the train is arriving at time t=6, then the train needs to arrive at platform #1.
  + If two or more trains are arriving at the same time then above two rules should be applied first for a train with smaller train number. This train's allocation will be decided first and the next qualified platform will be allocated to the other train. For example, consider the table below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Case No.** | **Platform #1 free since** | **Platform #2 free since** | **Train number 1 arriving at** | **Train number 2 arriving at** | **Resulting allocation** |
| Case 1 | t=5 | t=6 | t=7 | t=7 | Train 1 is allocated platform #1 and Train 2 is allocated platform #2 |
| Case 2 | t=5 | t=5 | t=7 | t=7 | Train 1 is allocated platform #1 and Train 2 is allocated platform #2 |
| Case 3 | t=6 | t=5 | t=7 | t=7 | Train 1 is allocated platform #2 and Train 2 is allocated platform #1 |

Constraints

1 <= N <= 5.10^4

0 <= A <= 86400

0 < I <= 86400

1000 < T < 10^8

Number of platforms > 0

Input

First line contains an integer N denoting number of trains.

Next N lines contain three integers T, A and I denoting the train number(T), the arrival time(A) and stoppage interval(I) of train respectively.

Next line contains an integer F denoting the train number for whom the allocated platform is to be reported in the output.

Output

First line contains an integer denoting the platform number on which the train F arrived.

Second line contains an integer denoting the busiest platform number on which the maximum trains arrived.

**OR**

If there are more than one platform having same traffic, sort all such platforms and print them in ascending order with each platform number on a new line. For example, if platform #2 and #3 are the busiest platforms then print as given below -

2

3

Time Limit

2

Examples

Example 1

Input

3

12121 15 5

12311 5 10

17215 2 3

12311

Output

2

1

Explanation

The earliest arriving train (17215) is at time t = 2 will arrive at platform# 1. Since it will stay there till t = 5, train (12311) arriving at time t = 5 will arrive at platform# 2 as the previous train (17215) will leave at t=5. Since current train (12311) will depart at time t = 15, next train (12121) arriving at time t = 15 will arrive at platform# 1 as the previous train (17215) left at t=5.

Here, maximum traffic came on platform #1. And train 12311 arrived at platform #2.

Example 2

Input

4

12106 2 4

12144 7 2

02812 3 2

13411 5 5

13411

Output

3

2

Explanation

The earliest arriving train (12106) at time t = 2 will arrive at platform# 1. Since it will stay there till t = 6, train (02812) arriving at time t = 3 will arrive at platform# 2. Since it will stay there till t = 5, train (13411) arriving at time t = 5 will arrive at platform# 3 Since it will stay there till t = 10, train (12144) arriving at time t = 7 will arrive at platform #2 as the previous train (02812) left at t=5.

Here, maximum traffic came on platform #2. And train 13411 arrived at platform #3

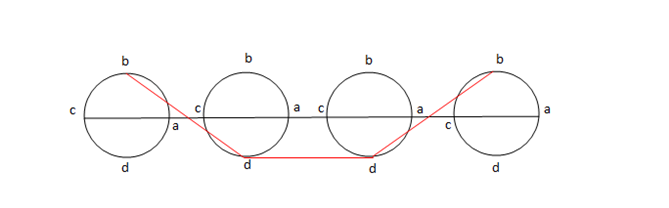
## Jogging Ground

### Problem Description

There are 4 circular grounds of equal sizes. Their circumferences do not intersect. The radius and the distance of the center of each circle from the leftmost circle's center are given.

There are 4 joggers who can start at the same time from any of the points designated as { a, b, c, d } on the circumference of all the four circles as shown in the diagram below. All 4 joggers jog in different grounds along the circumference of that ground. They could jog in either clockwise (left to right) or anticlockwise (right to left) direction. Finally they may also jog at different speeds.

Given starting position, direction of jogging and speed of jogging of all the 4 joggers, find the summation of length of 3 segments between the four joggers at a given point in time since the start of the jog.



Note: All the computation has to be accurate up to 6 digits after the decimal point.

### Constraints

1 <= N < 10^9

### Input

First line contains 4 integers each denoting the following

* R denotes the radius of all four circles
* D1 denotes the distance centre of the second circle from left to the centre of the leftmost circle
* D2 denotes the distance centre of the third circle from left to the centre of the leftmost circle
* D3 denotes the distance centre of the last circle from left to the centre of the leftmost circle

Second line contains 4 space separated integers denoting the angle with point a of each of the 4 circles where 0 degree indicates point a itself, 90 degree indicates point b, 180 degree indicates point c and 270 degree indicates point d.

Third line contains 4 space separated integers denoting the velocity in degrees per second.

Fourth line contains 4 space separated integers denoting the direction of running for joggers (0=clockwise and 1=anticlockwise).

Fifth Line contains integer N denoting the time in seconds since the start of the jog.

### Output

Print the summation of length of 3 segments between the four joggers after N seconds, rounded to the nearest integer.

### Time Limit

1

### Examples

Example 1

Input

10 25 50 75

0 0 0 0

1 1 1 1

1 1 1 1

90

Output

75

Explanation

Here every jogger is starting from point a and all have speed of 1 degree per second. So they will be at 90 degree after 90 seconds. After connecting these points we get segment lengths as (25 +25 +25) = 75

Example 2

Input

10 25 50 75

0 0 0 0

1 2 3 4

0 0 0 0

90

Output

91

Explanation

Here every jogger is starting from point a. They are jogging at the speed of 1, 2, 3 and 4 degrees per second respectively in clockwise direction. Hence after 90 seconds they will reach points where the segment length between them is (18.027800+36.400500+36.400500) = 90.8288. Hence, output is 91

## Max Funds

### Problem Description

An NGO wants to arrange funds for flood relief. It has divided volunteers into groups. A volunteer can only be a part of single group. Your task is to find the maximum funds collected by a group.

You will be given the funds collected by each volunteer and grouping pairs of the volunteers. You need to group the volunteers through these pairs.

### Constraints

0 < N, P <= 10000

0 < A, B <= N

### Input

First line contains one integer N, denoting number of volunteers.

Second line contains N space separated integers, representing the amount collected by each volunteer. The index of integer is the volunteer number starting from 1.

Third line contains the number of pairs, P.

Next P lines contain two space separated integers, A and B where A represents the first person and B represents the second person in the pair.

### Output

One line containing an integer, representing the maximum funds collected by the group.

### Time Limit

2

### Examples

Example 1

Input

5

23 43 123 54 2

3

1 3

2 3

1 2

Output

189

Explanation

In the above example, we have five volunteer [1, 2, 3, 4, 5] who have collected [23, 43, 123, 54, 2] respectively.

We have three groups that consists of [1, 2, 3], [4], [5]. First group collects 189 units of money, second group collects 54 units of money and third group collects 2 units of money. The maximum funds collected by any group is 189. Hence, the output is 189

Example 2

Input

9

34 54 65 76 88 23 56 76 43

7

1 3

2 3

1 2

6 8

5 4

5 7

8 9

Output

220

Explanation

In the above example, we have three groups that consists of [1, 2, 3], [4, 5, 7], [6, 8, 9]. Each group collects 153, 220, 142 units of money respectively. The maximum funds collected by a group is 220. Hence the output is 220.

## Minimum Penalty

### Problem Description

You are playing a game in which you have to beat n players. You are invincible; hence you are always able to beat your opponent. After beating a player, you will get some points. Your task is to get maximum score by beating them all. You can pick anyone randomly for maximizing your score. Rules of the game are as follows:

* It is guaranteed that you will be able to beat your opponent, hence your real task is to choose the best order to achieve maximum final score.
* When you beat a player, you get some points and a score will be generated.
* Here are some rules related to the game and maximizing your score:
  + All your opponent players are standing in one line next to each other i.e. the order of opponent is fixed
  + Your task is to choose a suitable opponent from this line
  + When you choose one opponent from that line, he steps out of the line and competes with you
  + After you beat him, you get to decide how your score for winning against him will be calculated
  + Essentially, if the opponent you have beaten has two neighbours, then you have the option to multiply the opponent number with any one of the two neighbours and subtract the other opponent number. That value becomes your winning score for that match
  + If your opponent has only one neighbor then your winning score for that match is product of current opponent number with neighbours opponent number
  + When dealing with last opponent in the game, your winning score is equal to the value of the last opponent number
  + As the game proceeds, the opponent that you have beaten has to leave the game

Example: 2 5 6 7

This depicts that you have four opponents with numbers 2 5 6 and 7 respectively

1. Suppose you choose to play with opponent number 5, then after winning, the max score for this match = 5\*6 - 2 = 28

Now opponent number 5 is out of the game. So opponent numbers 2 6 7 remain

1. Suppose you now choose to play with opponent number 2, then after winning, the max score for this match = 2\*6 = 12. Your overall score is now 28 + 12 = 40

Now opponent number 2 is out of the game. So opponent number 6 7 remain

1. Suppose you now choose to fight opponent number 6, then after winning, the max score for this match = 6\*7 = 42. Your overall score is now 40 + 42 = 82

Now opponent number 6 is out of the game. So opponent number 7 remains

1. After beating opponent number 7, the max score for this match is 7

So overall score in this case is 89. Hence when the order is 5->2->6->7 the overall score is 89.

Some other orders of choosing opponents will yield the following overall score

* Order 7->2->6->5 will yield overall score as 87
* Order 2->5->6->7 will yield overall score as 89
* Order 5->6->2->7 will yield overall score as 89
* Order 6->7->2->5 will yield overall score as 87
* But by following the order 6->5->2->7 will yield overall score as 91, which is maximum.

Hence, the correct answer yielded by order 6->5->2->7 is 91.

Your task is to maximize your score by taking the right decisions

### Constraints

1 <= N <= 500

0 <= individual points of opponents < 100

### Input

First line contains an integer N which denotes the number of opponents in the game

Second line contains N space separated integers, which are the opponent numbers of other opponents

### Output

Print the maximum obtainable score by making all the right decisions

### Time Limit

1

### Examples

Example 1

Input

4

2 5 6 7

Output

91

Explanation:

Refer the explanation in problem description.

Example 2

Input

3

7 8 9

Output

137

Explanation:

1. You choose to play with opponent number 8, then after winning, the max score for this match is = 8\*9-7 = 65

Now opponent number 8 is out of the game. So opponent numbers 7 9 remain

1. Suppose you now choose to play with opponent number 7, then after winning, the max score for this match is = 7\*9 = 63. Your overall score is now 65 + 63 = 128

Now opponent number 7 is out of the game. So opponent number 9 remains

1. After beating opponent number 9, the max score for this match is = 9

So overall score in this case is 128 + 9 = 137.

There is no other optimal order in which final score can be better than 137.

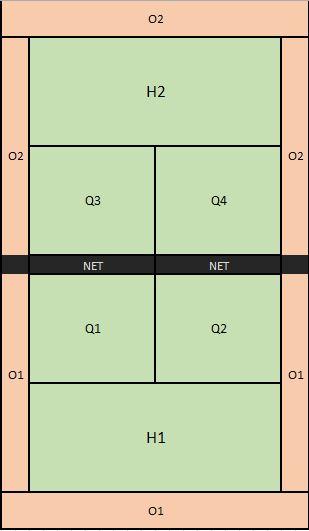
## Tennis Score

### Problem Description

A game of tennis (singles) is played. Regions of the court are named as shown in the image.

The running score of each game is described in the manner: "0", "15", "30", and "40". If at least three points have been scored by each side and a player has one more point than his opponent, the score of the game is advantage ("Advantage") for the player in the lead. (source: Wikipedia)

A set consists of a sequence of games played with service alternating between games, ending when a player wins a set by winning at least six games and at least two games more than the opponent. If one player has won six games and the opponent five, an additional game is played. If the leading player wins that game, the player wins the set 7-5. If the trailing player wins the game (tying the set 6-6) a tie-break is played. A tie-break, allows one player to win one more game and thus the set, to give a final set score of 7-6 or 6-7.(source: Wikipedia)



Aim is to toss the ball on the green part of the other side of the court. A set of strings will be provided signifying where the tennis ball has been tossed on the court. Assume that no player will hit the ball directly (i.e. without the ball being tossed on his (green) side of the court).

For Example, a string Q4 Q1 Q3 O1 will mean that server serves and the ball hits the area Q4. Receiver returns and ball hits ground in area Q1, server hits back again to area Q3 and then the receiver hits 'long' to area O1 and loses a point. So the set output is 0-0 and the current game output will be 15-0.

Following are the rules of the game:

* Game always starts with Server on H1 side (lower side in the image)
* Serve changes after every game is won
* On 'Serve' ball should hit on any 'Q' part of the other side. Hitting on 'H' part will be considered a fault
* While serving, if the server makes a double fault (ball should fall on his side or outside region twice), the server loses one point
* Points scored by the current server are mentioned first, for example if server wins the first point score will be 15-0
* At the end of a set, a changeover happens i.e. players change sides of the court.
* In case of game score 40-40, display "Deuce". In case of Server's Advantage, display "Advantage Server".

In case of Receiver's Advantage, display "Advantage Receiver"

* Number of sets played may not exceed 5
* In case a set is complete a set score of 7-5 will be denoted as:

7 0 (first player set score)

5 0 (second player set score)

Since the second set is about to start, a score of 0 0 is displayed. Please read these scores vertically.

* When a new game is about to start, display 0 - 0 for new game. For example:

0 0 (current game score)

**Departure from Tennis rule:**

* In real game of tennis a server is required to serve *cross court.*However, in this problem the server can serve in any court {Q3, Q4} for server 1 and {Q1, Q2} for server 2
* In real game of tennis a tie-break is counted in scores of 1, 2, 3, 4 instead of 15, 30, 40 etc. However, in this problem a tie-break is won according to regular rule i.e. by scoring points like in a regular game. For example, lets say the score is 6-6. In regular tennis a tie break would follow, but in this problem the 13th game will be played and scored using the same rules applied for first 12 games. Whoever wins the 13th game, wins the set i.e. either 7-6 or 6-7
* In a real game of tennis, a *changeover*(players switching court sides) happens at the end of a set or after the first game is played in a new set. However, in this problem changeover happens at the end of the set

### Constraints

Number of space separated strings in input < 500

### Input

One line containing a string representing the sequence where the ball drops, separated by space

### Output

First line containing the set score of all the sets of the Server (starting from first and separated by space)

Second line containing the set score of all the sets of the Receiver (starting from first and separated by space)

Third line containing the game score of the current game (separated by space) or "Deuce" or "Advantage Receiver" or "Advantage Server", as may be the case.

### Time Limit

1

### Examples

Example 1

Input

Q1 Q1

Output

0  
0  
0 15

Explanation

Server serves and ball hits area Q1. He will not lose point because he is serving. Again he serves and ball hits area Q1. Since, this is a double fault he loses one point. Hence the game score is 0-15.

Example 2

Input

Q1 Q1 Q3 Q3 Q1 Q1 Q3 Q3 Q1 Q1 Q3 Q3

Output

0  
0  
Deuce

Explanation

· String Q1 Q1 indicates a double fault. The score becomes 0-15

· Next String Q3 indicates a valid serve. The second Q3 indicates that the Receiver could not return the serve because it fell into his own half. This makes the score 15-15.

· Similar sequence repeats for two more times, making the score 30-30, then 40-40 which is Deuce.

Example 3

Input

Q1 Q1 Q2 Q3 O1 H2 H2 Q1 Q1 Q1 Q2

Output

1  
0  
0 0

Explanation

· Except points Q2 Q3 O1 all other string indicates a double fault on part of the server

· Q2 Q3 O1 earns the server one point

· Overall the server loses the game

· Now a new game begins and the current server becomes receiver and vice versa

· **The score is printed after the new game begins** and the score is 0 0. Hence, server score is 1 whereas the receiver score is 0