1. Count Pairs

Problem Description

Given an array of integers A, and an integer K find number of happy elements.

Element X is happy if there exists at least 1 element whose difference is less than K i.e. an element X is happy, if there is another element in the range [X-K, X+K] other than X itself.

Constraints

1 <= N <= 10^5

0 <= K <= 10^5

0 <= A[i] <= 10^9

Input

First line contains two integers N and K where N is size of the array and K is a number as described above

Second line contains N integers separated by space.

Output

Print a single integer denoting the total number of happy elements.

Time Limit

1

Examples

Example 1

Input

6 3

5 5 7 9 15 2

Output

5

Explanation

Other than number 15, everyone has at least 1 element in the range [X-3, X+3]. Hence they are all happy elements. Since these five are in number, the output is 5.

Example 2

Input

3 2

1 3 5

Output

3

Explanation

All numbers have at least 1 element in the range [X-2, X+2]. Hence they are all happy elements. Since these three are in number, the output is 3.

2. Signal Connection

Problem Description

You have been given longitude and latitude of locations from where the channels are going to be broadcasted. You also get the height of tower from which these channels are broadcasted. Moreover, you have been given the location of your friend Jason. You have to calculate how many connections he can make to the towers. Take Radius of earth= 6371 KM.

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

Constraints

1 <= N < 10^5

Input

First line contains integer N denoting the number of locations from where the channel is going to be broadcasted

Second line contains N space-separated decimal values denoting latitudes

Third line contains N space-separated decimal values denoting longitudes

Fourth line contains N space-separated integer values denoting the height of tower from which channels are broadcasted

Fifth Line contains two space-separated decimal values denoting latitude, longitude of Jason's location

Output

Print the number of channels Jason can connect with

Time Limit

1

Examples

Example 1

Input

2

19.076090 17.387140

72.877426 78.491684

2 1

18.516726 73.856255

Output

1

Explanation

First latitude and longitude is Mumbai and second is for Hyderabad from where the channel signals are broadcasted. Jason is in Pune. According to signal strength Jason will only be able to connect Mumbai tower.

Example 2

Input

2

28.644800 22.572645

77.216721 88.363892

5 7

48.864716 2.349014

Output

0

Explanation

First latitude and longitude is Delhi and second is for Kolkata from where the channel signals are broadcasted. Jason is in Paris. According to signal strength Jason will not be able to connect any tower.

3. Railway Station

Problem Description

Given schedule of trains and their stoppage time at a Railway Station, find minimum number of platforms needed.

Note -

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.

Constraints

1 <= N <= 10^5

0 <= a <= 86400

0 < b <= 86400

Number of platforms > 0

Input

First line contains N denoting number of trains.

Next N line contain 2 integers, a and b, denoting the arrival time and stoppage time of train.

Output

Single integer denoting the minimum numbers of platforms needed to accommodate every train.

Time Limit

1

Examples

Example 1

Input

3

10 2

5 10

13 5

Output

2

Explanation

The earliest arriving train at time t = 5 will arrive at platform# 1. Since it will stay there till t = 15, train arriving at time t = 10 will arrive at platform# 2. Since it will depart at time t = 12, train arriving at time t = 13 will arrive at platform# 2.

Example 2

Input

2

2 4

6 2

Output

2

Explanation

Platform #1 can accommodate train 1.

Platform #2 can accommodate train 2.

Note that the departure of train 1 is same as arrival of train 2, i.e. 6, and thus we need a separate platform to accommodate train 2.

4. Best Sequence

Problem Description

Some of the keys of Ajith's Laptop's Keyboard are damaged and he is not able to type those keys. He has to complete his assignment and submit it the next day and since it is midnight he will not be able to give his laptop for repair. So he decides to make a character sequence of **all** the damaged keys in a sequence that he can copy and paste and make a word out of them.
Ajith needs to type a paragraph with all the characters in lower case. Help Ajith to find out the best permutation of the sequence of the characters (corresponding to the damaged keys) per word, that can be used while typing the paragraph, i.e. the sequence that will require least insertion and deletion while typing a word. Consider paste operation to be of one keystroke. Ignore the copy operation.

Recursively apply the same procedure for all the words in the paragraph. This way you will get the best combination that should be selected for that word. Finally, how many different words exist per character sequence combination. The combination that is the best for maximum words should be printed as output. If there are more than one candidates for best character sequence print the lexicographically smallest character sequence.

Refer the example section for better understanding.

Constraints

0 < Number of words in paragraph < 50

0 < Number of damaged keys <= 6

Input

First line contains the paragraph P that is to be written Second line contains the characters that represent the damaged keys

Output

One Line containing the best string which can be used to copy paste for the words.

Time Limit

1

Examples

Example 1

Input

supreme court is the highest judicial court

s u

Output

su

Explanation

There are two possible combinations of the damaged keys i.e. either *su* or *us*.
For word **supreme**, *su* is suitable as it requires only paste operation
For **court**, *us* is suitable as it requires less keystrokes for deletion
Similarly, for **is**, *su* is suitable and so on
Finally, we get *su* suitable for words supreme, is and highest and *us* suitable for words court (twice) and judicial. We get *su* and *us* suitable for 3 words each.
Since *su* is lexicographically smaller than us. So, the output will be *su*.

Example 2

Input

ginnestinggin gniinginging

n i g

Output

gin

Explanation

There are six possible combinations of the damaged keys i.e. {*nig, ngi, ing, ign, gni, gin*}.
For the first word, *gin* sequence requires least keystrokes for insertion and deletion. Similarly, for the second word *ing* sequence requires least keystrokes. Since both are eligible for best sequence, we will need to output the lexicographically smaller string. So, the output will be *gin*.

5. Hedger

Problem Description

A hedger is an investor who takes steps to reduce the risks of investment by doing appropriate research and analysis of stocks. Assume that you are a hedger.

Now you have been given some parameters based on which you have to analyze and pick the correct stocks. You have been assigned a fund that you have to manage in such a way that it should give maximum returns.

According to your research you have a list of stocks for which you know the corresponding profit percentages that you can earn within a certain horizon. Being a hedger you don't want to put all your eggs in one basket. That's why you have decided the upper limit in terms of quantity that you will buy.

Given number of stocks, the upper limit on quantity, the amount to be invested, the list of stock prices and list of profit percentages, calculate the maximum profit you can make.

Note: All computation should be up to two digits after the decimal point.

Constraints

0 <= A <= 10^6

1 <= K <= 100

1 <= N <= 10^4

Input

First line contains three space separated integers N K A where N is number of stocks in Market, K is maximum quantity of any particular stock you can buy and A is capital amount you have.

Second line contains N space separated decimals denoting the prices of stocks.

Third line contains N space separated decimals denoting the profit percentages corresponding to the index of stock prices.

Output

Print the maximum profit that can be earned from the given amount, rounded to the nearest integer.

Time Limit

1

Examples

Example 1

Input

4 2 100

20 10 30 40

5 10 30 20

Output

26

Explanation

Here, we can select only two stocks of any stocks that we choose to buy. We choose to buy stocks which are priced at 30 and 40 respectively. Before we exhaust our capital maximum profit that can be earned

Profit=2*((30*30)/100) +1*((40*20)/100)=26

Example 2

Input

5 3 200

90 25.5 15.5 30.8 18.8

5 10 20 5.5 2.5

Output

20

Explanation

Here, we can select only three stocks of any stocks that we choose to buy. We choose to buy stocks which are priced at 25.5, 15.5 and 30.8 respectively. Before we exhaust our capital maximum profit that can be earned

Profit=3*((25.5*10)/100) +3*((15.5*20)/100)+2*((30.8*5.5)/100)=20.34 Hence, output is 20.

6. Maximum Prize

Problem Description

Imagine you are a martial arts fighter fighting with fellow martial artists to win prize money. However unlike traditional competitions, here you have the opportunity to pick and choose your opponent to maximize your prize kitty. The rules of maximization of prize kitty are as follows

► You have a superpower bestowed upon you, that you will win against anyone you challenge

► You have to choose the right order because unfortunately the superpower does not ensure that your prize money is always the highest

► Every victory against an opponent that you challenge and win against, will translate into a certain winning sum

► Here begins the technical part that you need to know in order to maximize your winning prize money

- All your opponents are standing in one line next to each other i.e. the order of opponents is fixed
- Your first 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 fights you.
- After you beat him, you get to decide how your prize money 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 add the other *opponent number.* That value becomes your prize money for that match
- If your opponent has only one neighbor then your prize money for that match is product of current *opponent number* with neighbours' *opponent number*
- When dealing with last opponent in the tournament, your prize money is equal to the value of the last *opponent number*
- As the tournament proceeds, the opponent that you have beaten has to leave the tournament

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 fight opponent number 5, then after winning, the max prize kitty you can win for that match is = 5*6+2 = 32

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

2. Suppose you now choose to fight opponent number 2, then after winning, the max prize kitty you can win for that match is = 2*6+0 = 12. Your overall prize kitty is now 32 + 12 = 44

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

3. Suppose you now choose to fight opponent number 6, then after winning, the max prize kitty you can win for that match is = 7*6+0 = 42. Your overall prize kitty is now 44 + 42 = 86

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

4. After beating opponent number 7, the max prize kitty you can win for that match is 7

   So overall prize kitty in this case is 93.

Other orders of choosing opponents will yield the following overall prize kitty

- Order 7->2->6->5 will yield overall prize kitty as 87
- Order 2->5->6->7 will yield overall prize kitty as 88
- Order 5->6->2->7 will yield overall prize kitty as 95
- Order 6- >7->2->5 will yield overall prize kitty as 97
- But by following the order 6->5->2->7 will yield overall prize kitty as 105, which is maximum.

Your task is to maximize your prize kitty by taking the right decisions

Input

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

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

Output

Print the maximum number of coins you can win

Constraints

1 <= N <= 500

0 <= individual coin count < 100

Time Limit

1

Examples

Example 1

Input

4

2 5 6 7

Output

105

Explanation:

Refer the explanation in problem description.

Example 2

Input

3

7 8 9

Output

151

Explanation:

1. You choose to fight opponent number 8, then after winning, the max prize kitty you can win for that match is = 8*9+7 = 79

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

2. Suppose you now choose to fight opponent number 7, then after winning, the max prize kitty you can win for that match is = 7*9+0 = 63. Your overall prize kitty is now 79 + 63 = 142

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

3. After beating opponent number 9, the max prize kitty you can win for that match is 9

So overall prize kitty in this case is 142 + 9 = 151.