Fill The Cube

Problem Description

A company manufactures walls which can be directly implanted at the site. The company uses small square bricks of material C and material D which have similar looks but have huge difference in quality. The company manufactures walls of square shapes only to optimize their costs.

A novice employee created a square wall using bricks of material C and D. However, the client had asked the wall to be made of only high-quality material - material C.

To solve this problem, they will place the wall in a special furnace and heat it such that the material D melts and only material C remains. Material C brick will move down due to gravity if a material D brick below it melts. The new empty space created will be filled by new material C square walls. They also want to use biggest possible C square wall while building the final wall. For this they will position the wall in the furnace in an optimal way i.e. rotate by 90-degrees any number of times, if required, such that the biggest space possible for new material C wall is created. No rotations are possible when the furnace starts heating.

Given the structure of the original wall created by the novice employee, you need to find out the size of the new C square wall which can be fitted in the final wall which will be delivered to the client.

Constraints

$1 < N < 100$

Input

First Line will provide the size of the original wall N.

Next N lines will provide the type of material (C and D) used for each brick by the novice employee.

Output

Size of the biggest possible C square wall which can be fitted in the final wall.

Time Limit

1

Examples

Example 1

Input

4

C D C D

C C D C

D D D D

C D D D

Output

3

Explanation

If the wall is placed with its left side at the bottom, space for a new C wall of size 2x2 can be created. This can be visualized as follows

D C D D

C D D D

D C D D

C C D C

The melted bricks can be visualized as follows

- - - -

- C - -

C C - -

C C - C

Hence, the maximum wall size that can be replaced is 2x2.

If the wall is placed as it is with its original bottom side at the bottom, space for a new C wall of size 3x3 can be created. Post melting, this can be visualized as follows.

- - - -

C - - -

C - - -

C C C C

Hence, the maximum wall size that can be replaced is 3x3 in this approach.

Since no rotations followed by heating is going to a yield a space greater than 3x3, the output is 3.

Example 2

Input

7

C D D C D D D

C D D C D D D

D D D D D D C

D C D C D D D

D D D C D C D

C D D C D C C

C D C D C C C

Output

5

Explanation

If the wall is placed with its left side at the bottom, a space for new C wall of size 5x5 can be created. This can be visualized as follows

D D C D D C C

D D D D C C C

D D D D D D C

C C D C C C D

D D D D D D C

D D D C D D D

C C D D D C C

When this orientation of the wall is heated, a space for new C wall of size 5x5 is created after the D bricks melt

_ _ _ _ _ _ _

_ _ _ _ _ _ _

_ _ _ _ _ _C

_ _ _ _ _ C C

_ _ _ _ _ C C

C C _ C C C C

C C C C C C C

Whereas, if the rotation was not done, the wall formed after the D bricks melt will be as follows

_ _ _ _ _ _ _

_ _ _ _ _ _ _

_ _ _ C _ _ _

C _ _ C _ _ _

C _ _ C _ _ C

C _ _ C _ C C

C C C C C C C

When this orientation of the wall is heated, a space for new C wall of size 3x3 only is created after the D bricks melt

Hence rotation is important and correct answer is 5x5

Since no rotations followed by heating is going to a yield a space greater than 5x5, the output is 5

Max Sum

Problem Description

Two friends A and B are playing with an array of integers. They both agree upon the operations to be performed on the array but differ on choice of window size to perform the said operations.

The operations to be performed on the array are explained below:

· One can choose to add at the most X consecutive elements.

· After performing the addition operation, it is mandatory to skip the next element in the array.

· The goal is to achieve maximum sum by choosing appropriate elements in the array.

A wants X to be W, while B wants X to be (W + D). This is the only difference that they have. Your task is to find out who wins. Winner is the person whose sum is higher.

The inputs that will be provided are values of elements of array, value of W and value of D.

Example:

Array: 4 5 6 1 2 7 8 9

Window Size (W): 3

Output: 39

Explanation

· We will choose to add elements 4, 5, 6 since window size is 3.

· Since one addition operation is complete, we have to skip one element which is 1.

· We choose to skip element 2 because the next three values are also higher than 2.

· The max sum thus obtained is 39.

Now suppose the array was: 4 5 6 1 2 3 7 8 9

· We will choose to add elements 4, 5, 6 since window size is 3.

· Since one addition operation is complete, we have to skip one element which is 1.

· Now we choose to pick element 2 because we can skip element 3 and still pick up the next 3 values viz 7, 8, 9.

· The max sum thus obtained is 41.

· Note that we picked up only one element in second selection since constraint is only on maximum number to be chosen, not minimum.

Now suppose the array was: 4 5 6 7

· Since one can start from any index, we choose element 5, 6, 7.

· The max sum thus obtained is 18.

The above examples illustrate the game with a fixed window size of W. Since B prefers to play the same game with the size of W+D, the steps will remain the same but the max sum output may be different. Print different output depending on whether A wins, B wins or it's a tie.

Constraints

0 <= N <= 10 ^ 5

5 <= W <= 10 ^ 5

-10^5 <= D <= 10^5

0 < (W + D) <= N

0 <= elements in array <= 10 ^ 9

Input

First line contains three space separated integers N and W and D respectively, which denote

N - size of array

W - window size

D - difference

Second line contains of N space separated integers denoting the elements of the array

Output

If B wins, print “Right <absolute difference>”

If A wins, print “Wrong <absolute difference>”

If It’s a tie, print “Both are Right”

Refer Examples section for better understanding.

Time Limit

1

Examples

Example 1

Input

8 5 -2

4 5 6 1 2 7 8 9

Output

Wrong 2

Explanation

Here we have given N = 8, W = 5, D = -2

A will maximize the sum of elements of the array using window size 5. Whereas B will maximize the sum of elements of the array using window size 3 (5-2).

Using logic as depicted above A will get the max sum as 41 and B will get the max sum as 39. The absolute difference is 41 - 39 = 2.

Hence, output will be: Wrong 2

Example 2

Input

9 2 2

4 5 6 1 2 3 7 8 9

Output

Right 10

Explanation

Here we have given N = 9, W = 2, D = 2

A will maximize the sum of elements of the array using window size 2. Whereas B will maximize the sum of elements of the array using window size 4 (2+2).

Using logic as depicted above A will get the max sum as 33 and B will get the max sum as 43. The absolute difference is 43 - 33 = 10.

Hence, output will be: Right 10

Example 3

Input

10 9 -3

4 5 6 3 2 3 7 8 9 2

Output

Both are right

Explanation

Here we have given N = 10, W = 9, D = -3

A will maximize the sum of elements of the array using window size 9. Whereas B will maximize the sum of elements of the array using window size 6 (9-3).

Using logic as depicted above A will get the max sum as 47 and B will get the max sum as 47. The absolute difference is 47 - 47 = 0.

Hence, output will be: Both are right

Largest Gold Ingot

Problem Description

Ramesh is a goldsmith, who brought a large number of gold ingot each of different length(L) but equal breadth(B) and height(H). He wants to weld the ingots of same length with each other. He tasks his new employee, Akash, to weld the ingots of same length with each other. But Akash forgot that he had to weld the ingots of same length, instead he welded the ingots in a random manner.

Later Ramesh found out what he had done. He then ordered Akash to cut the welded ingot such that a cuboid with the largest volume from the welded gold ingot is obtained.

Find the volume of summation of gold ingots minus volume of the largest cuboid.

Constraints

0 < G < 10^5

Input

First Line contains one integer G, denoting number of gold ingots

Second line contains two space separated integers B and H, where B denotes the breadth and H denotes the height of individual ingot

Third line contains G space separated integers, denoting the length of the individual gold ingots that are welded together in adjacent manner

Output

An integer corresponding to the volume of summation of gold ingots minus volume of the largest cuboid, mod 10^9+7.

Time Limit

1

Examples

Example 1

Input

7

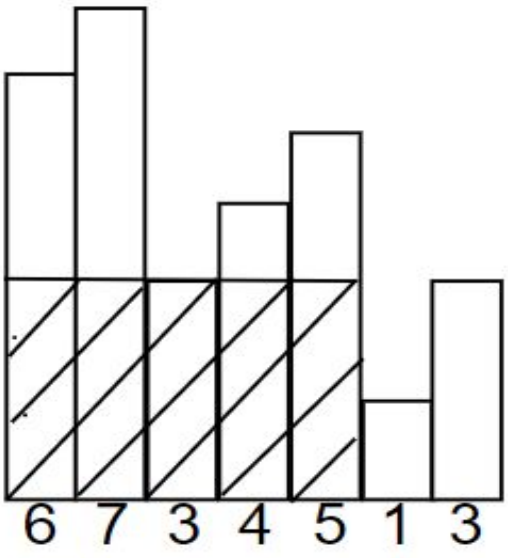
1 1

6 7 3 4 5 1 3

Output

14

Explanation



Total volume of shaded region is 15 and the total volume is 29. So the volume of summation of gold ingots minus largest cuboid obtained is 14, since the height is 1 and breadth is 1.

Example 2

Input

7

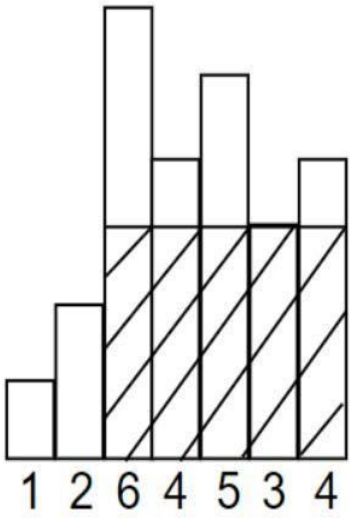
1 2

1 2 6 4 5 3 4

Output

20

Explanation



The volume of summation of gold ingots minus largest cuboid obtained is 20, since the height is 2 and breadth is 1.

Binary Equivalent

Problem Description

Mr. Binary is lost and wants to be found but the problem is he understands only binary. His house is located at a maximum binary equivalence possible, from the given set of numbers. A set is a binary equivalence if the number of 0 zeros and ones from a set of number are equal.

Constraints

1 <= N <= 20

1 <= Arr[i] <= 10^5, where Arr[i] is the ith element in the set of N numbers in second line of input

Arr[i] will be unique

Input

First line contains N denoting the number of decimal numbers

Next line contains N space separated decimal numbers

Output

Single line output printing possible binary equivalence where number of digits in this number is equal to number of bits present in the largest element in second line of input. If there is no set which has binary equivalence then return 0 padded to number of bits present in the largest element in second line of input.

Time Limit

1

Examples

Example 1

Input

3

2 7 10

Output

0011

Explanation

2 -> 0010 - 1's = 1, 0's = 3

7 -> 0111 - 1's = 3, 0's = 1

10 -> 1010 - 1's = 2, 0's = 2

Here we have taken up to 4 bits because the maximum number is 10 which needs 4 bits to be represented in binary. The number of zeroes and ones across the set is, 6

each. Hence, the set of [2,7,10] has binary equivalence. Similarly, if you consider set[2,7], it also has binary equivalence, 4 each. But set [7,10] does not have binary equivalence. Likewise, set[10] has binary equivalence of 2 each.

Total number of unique sets where binary equivalence is possible from all combinations are 3 viz. Sets are [2,7,10], [2,7] and [10] which is the final answer. But as Mr. Binary only understands zeroes and ones, return the binary of 3.

Since 10 is the largest element in the input on line 2, the number of bits required to represent 10 in binary is 4. Hence output needs to be padded upto 4 digits. Since binary of 3 represented as a 4-digit number is 0011, the answer is 0011

Note

Do not consider empty subset

Example 2

Input

1

7

Output

000

Explanation

7 -> 111 - 1's = 3, 0's = 1

Since there is only one element in the set and it also does not have binary equivalence, the answer is 0. However, keeping output specifications in mind, the answer should be printed as 000 since the highest element in second line of input viz. 7 has 3 bits when represented in binary format.

Secret Word

Problem Description

A computer scientist has developed an encryption algorithm. This algorithm takes two inputs - one plain word and another, a key. Characteristics of inputs are as below.

Plain word: It is a string consisting of lowercase alphabets only.

Key: It is a set of pairs of strings consisting of lowercase alphabets only. For each pair, first string is the plain word and second string is its secret word. The characters of these secret words are jumbled but lengths of Plain Word and Secret Word are equal.

This algorithm finds the secret characters for each character in the inputted plain word by using the key. Then it combines all the secret characters in the same order to form a string called the secret word. Finally output this secret word. Below table shows how secret characters can be obtained from the key.

Examples

Plain Word(s) in Key	Secret Word(s) in Key	What can be inferred?	Explanation
a	b	Secret character of 'a' is 'b'	Only one character in both plan word and secret word
hello	apple	Secret character of 'l' is 'p'	Character 'l' is repeating two times in plain word and 'p' is repeating two times in secret word
egg	ill	Secret character of 'g' is 'l' and that of 'e' is 'i'	Character 'g' is repeating two times in plain word and 'l' is repeating two times in secret word. Then, only one character remaining in both plain word and secret word, i.e, 'e' and 'i'
axe echo	cow atom	Secret character of 'e' is 'o'	Secret word of "axe" is "cow" and that of "echo" is "atom". Character 'e' is common in plain words and 'o' is common in secret words

Your task is to help him in implementing the algorithm as a computer program.

Note: It is guaranteed that all characters in the given plain word can be converted to secret characters by using the given key.

Note: It is guaranteed that one plain text can be converted to only one encrypted text.

Constraints

1 <= P <= 52000

1 <= N <= 26

1 <= Length of a plain word in pair <= 50000

1 <= Length of a secret word in pair <= 50000

Length (plain word) == Length (secret word)

Input

First line contains string P denoting the plain text.

Second line contains an integer N denoting number of key pairs.

Next N lines, each contain two space separated strings denoting plain text and key.

Output

Print the encrypted word.

Time Limit

1

Examples

Example 1

Input

load

3

app lol

old tip

odd itt

Output

piot

Explanation

"load" is the plain word to be encrypted. Given Key contains 3 pairs of Plain word and Secret word combination. They are <"app", "lol">, <"old", "tip"> and <"odd", "itt">. From first pair, it's clear that the secret character of 'p' is 'l' and that of 'a' is 'o'. From third pair, it's clear that the secret character of 'd' is 't' and that of 'o' is 'i'. By using above findings, from second pair, it is clear that the secret character of 'l' is 'p'. Now we can build the secret word by replacing the characters of plain word by its corresponding secret characters as "piot".

Example 2

Input

a

1

a b

Output

b

Explanation

The word "a" is the plain word to be converted to secret word. The given key consists of only one plain word - secret word pair. i.e., <"a", "b">. From this, it is clear that the secret character of 'a' is 'b', since there is only one character in both secret and plain words. So, the final output is "b".