**Q:-1**

You are given two binary strings aa and bb of the same length. You can perform the following two operations on the string aa:

* Swap any two bits at indices ii and jj respectively (1≤i,j≤n1≤i,j≤n), the cost of this operation is |i−j||i−j|, that is, the absolute difference between ii and jj.
* Select any arbitrary index ii (1≤i≤n1≤i≤n) and flip (change 00 to 11 or 11 to 00) the bit at this index. The cost of this operation is 11.

Find the minimum cost to make the string aa equal to bb. It is not allowed to modify string bb.

**Input**

The first line contains a single integer nn (1≤n≤1061≤n≤106) — the length of the strings aa and bb.

The second and third lines contain strings aa and bb respectively.

Both strings aa and bb have length nn and contain only '0' and '1'.

**Output**

Output the minimum cost to make the string aa equal to bb.

**Examples**

**input**

**Copy**

3  
100  
001

**output**

**Copy**

2

**input**

**Copy**

4  
0101  
0011

**output**

**Copy**

1

**Note**

In the first example, one of the optimal solutions is to flip index 11 and index 33, the string aa changes in the following way: "100" →→ "000" →→ "001". The cost is 1+1=21+1=2.

The other optimal solution is to swap bits and indices 11 and 33, the string aa changes then "100" →→ "001", the cost is also |1−3|=2|1−3|=2.

In the second example, the optimal solution is to swap bits at indices 22 and 33, the string aa changes as "0101" →→ "0011". The cost is |2−3|=1|2−3|=1.

**Q:-3**

**Light Up the Bulbs**

Send Feedback

A bulb can be ‘ON’ or ‘OFF’. Mr. Navdeep got ‘n’ number of bulbs and their status, whether they are ‘ON’ or ‘OFF’. Their status is represented in a string of size ‘n’ consisting of 0’s and 1’s, where ‘0’ represents the bulb is in ‘OFF’ condition and ‘1’ represent the bulb is ‘ON’. Mr. Navdeep has been given the task to light up all the bulbs.

He can perform two operations.

First, chose any segment of bulbs and reverse them means chose any substring and reverse it. E.g. “0 110 001” -> “0 011 001”. Substring (1, 3) is reversed here. This operation will cost him Rs. ‘X’.

Second, chose any segment of bulbs and reverse their present condition. i.e. if the bulb is ‘ON’, make it ‘OFF’ and if it is ‘OFF’, make it ‘ON’. E.g. “0 011 001” -> “0 100 001”. Substring (1, 3) is complemented. This operation will cost him Rs. ‘Y’.

You need to help Mr. Navdeep that how much minimum amount it will require to make all the bulbs lightened. (or make all the characters as ‘1’ in the representation string)

**Input Format:**

The first line will contain three space separated integers: ‘n’, ‘X’, ‘Y’ denoting the number of bulbs, cost of first operation and cost of second operation respectively.

The second line contains a representation string of length ‘n’ consisting of 0’s and 1’s representing whether the bulb is ‘OFF’ or ‘ON’.

**Output Format:**

Print a single integer denoting the minimum cost required to light up all the bulbs.

**Constraints:**

1 <= n <= 3,00,000

0 <= X, Y <= 10^9

Time Limit: 1 second

**Sample Input:**

5 1 10

01000

**Sample Output:**

11

**Explanation:**

First, Reverse substring (0, 1): “01000” -> “10000”, COST = 1

Second, Invert substring (1, 4): “10000” -> “11111”, COST = 10

Total cost = 1+10 => 11

**Q:-4**

**Circular List of Students**

Send Feedback

You are given a circular list of students as follows:

0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10 -> 11

This list is circular, means that 11 will follow 0 again. You will be given the student number ‘i’ and some position ‘p’. You will have to tell that if the list will start from (i+1)th student, then which student will be at pth position.

**Input Format:**

First line will have an integer ‘t’, denoting the number of test cases.

Next line will have two space separated integers denoting the value of ‘i’ and ‘p’ respectively.

**Output Format:**

Print ‘t’ lines containing single integer denoting the student number.

**Constraints:**

1 <= t <= 10^5

0 <= i <= 11

1 <= p <= 12

**Sample Input:**

2

2 3

5 8

**Sample Output:**

5

1

**Explanation:**

First, list will start at 3. 3 -> 4 -> 5. Hence, 5 will be at third position.

Second, list will start at 6. 6 -> 7 -> 8 -> 9 -> 10 -> 11 -> 0 -> 1. Hence, 1 will be at 8th position.

**Q:-5**

**Interesting Sequences**

Send Feedback

Professor Jain has a class full of notorious students. To get anything done from them is a herculean task. Prof Jain wanted to organize a test. He gave this responsibility to Aahad. Aahad did an excellent job of organizing the test. As a reward, the professor gave him an sequence of numbers to play with. But Aahad likes playing with "interesting" sequence of numbers, which are sequences that have equal elements.

Now, the problem is - Prof Jain has a sequence with elements, and that sequence isn't always "interesting”. To ensure sequence has equal elements, Prof Jain has 2 options:

1) Choose two elements of sequence . DECREASE the first element by 1 and INCREASE the second element by 1. This operation costs 'k' coins.

2) Choose one element of array and INCREASE it by 1. This operation costs 'l' coins.

What’s the minimum number of coins Prof Jain needs to turn his sequence into a “interesting" sequence for Aahad?

**Input Format**

The first line of input contains three space-separated integers: n, k, l . Integer n is the size of array . Integer k is the number of coins needed to perform the first operation. Integer l is the number of coins needed to perform the second operation.

The second line contains n integers: (a1, a2, a3... an) representing sequence.

**Constraints:**

1 <= n, k, l <= 1000

1 <= ai <= 1000

Time Limit: 1 second

**Output Format**

In single line, print one integer number: the minimum number of coins required to make "interesting" sequence.

**Sample Test Cases:**

Sample Input 1:

4 1 2

3 4 2 2

Sample Output 1:

3

Explanation Output 1 :

The professor has a sequence with 4 elements. To perform the first operation, they must pay 1 coin and to perform the second operation, they must pay 2 coins. The optimal strategy is:

-Perform the second operation on the fourth element. Now the sequence is {3, 4, 2, 3}. This costs 2 coins.

-Perform the first operation on the second and third element. The sequence is now "interesting", and it looks like {3, 3, 3, 3}. This costs 1 coin.

The total amount of coins needed is 2 + 1 = 3.

Sample Input 2:

3 2 1

5 5 5

Sample Output 2:

0

Explanation Output 2 :

The given sequence is already "interesting". The professor would spend 0 coins.

Sample Input 3:

5 2 1

1 2 3 4 5

Sample Output 3:

6

Explanation Output 3 :

The professor has a sequence with 5 elements. To perform the first operation, they must pay 2 coin and to perform the second operation, they must pay 1 coin. The optimal strategy is:

-Perform the first operation on the first and last element. Now the sequence is {2, 2, 3, 4, 4}. This costs 2 coins.

-Perform the first operation again on the first and last element. Now the sequence is {3, 2, 3, 4, 3}. This costs 2 coins.

-Perform the first operation on the second and second last element. Now the sequence is {3, 3, 3, 3, 3}. This costs 2 coins.

The total amount of coins needed is 2 + 2 + 2 = 6.

**Q:-6**

**Winning strategy**

Send Feedback

Our college team is going to the sports fest to play a football match with our coach. There are n players in our team, numbered from 1 to n.

The coach will know the position of another team hence create a winning strategy. He creates the position of every player in a specific order so that we will win and then he starts swapping two players at a time to form the positions.

He swaps payers in such a way that it can't be understood by another team:

1. Any player can swap with the player directly at front him

2. One player can swap at most with two other players

If the specific order is formed then our team will win otherwise we will lose

**Input Format**

The First line contains numbers of players in team: n

The second line contains n space separated integers denoting the specific position of players: i-th integer denotes the position of Ai player in winning strategy

**Output Format**

If our team wins print "YES"(without quotes) and in next line print the minimum numbers of swapping required to form this specific order otherwise print "NO"(without quotes)

**Constraints**

1 =< n <= 10^5

1 <= Ai <= n

**Sample Input1:**

5

2 1 5 3 4

**Sample Output1:**

YES

3

**Sample Input2:**

5

2 5 1 3 4

**Sample Output3:**

NO

**Explaination**

In the First Sample case,

Initial state: 1 2 3 4 5

Three moves required to form this order:

1 2 3 4 5 -> 1 2 3 5 4 -> 1 2 5 3 4 -> 2 1 5 3 4

In the second case, no way to form this specific order