

Contest Discussion

1. Sum of all odd length Subarray.
2. Little Pony and Maximum Element.
3. Vowels in a Range.

1. Sum of all odd length Subarray

Problem Description

Given an array of positive integers A of size N, return the sum of all possible odd-length subarrays of A.

A subarray is a contiguous subsequence of the array.

Problem Constraints

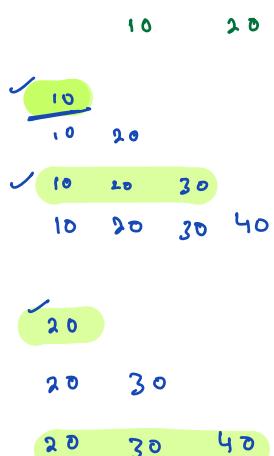
$$1 \leq N \leq 10^5 \rightarrow (N^2) \quad (10^5)^2 = 10^{10}$$
$$-10^9 \leq A[i] \leq 10^9$$

Input Format

Single argument which is an integer array A.

Output Format

Return an integer.

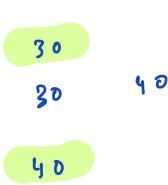
E x 1 :

$$\frac{n(n+1)}{2} = \frac{2 \times 5}{2}$$

$$= 10$$

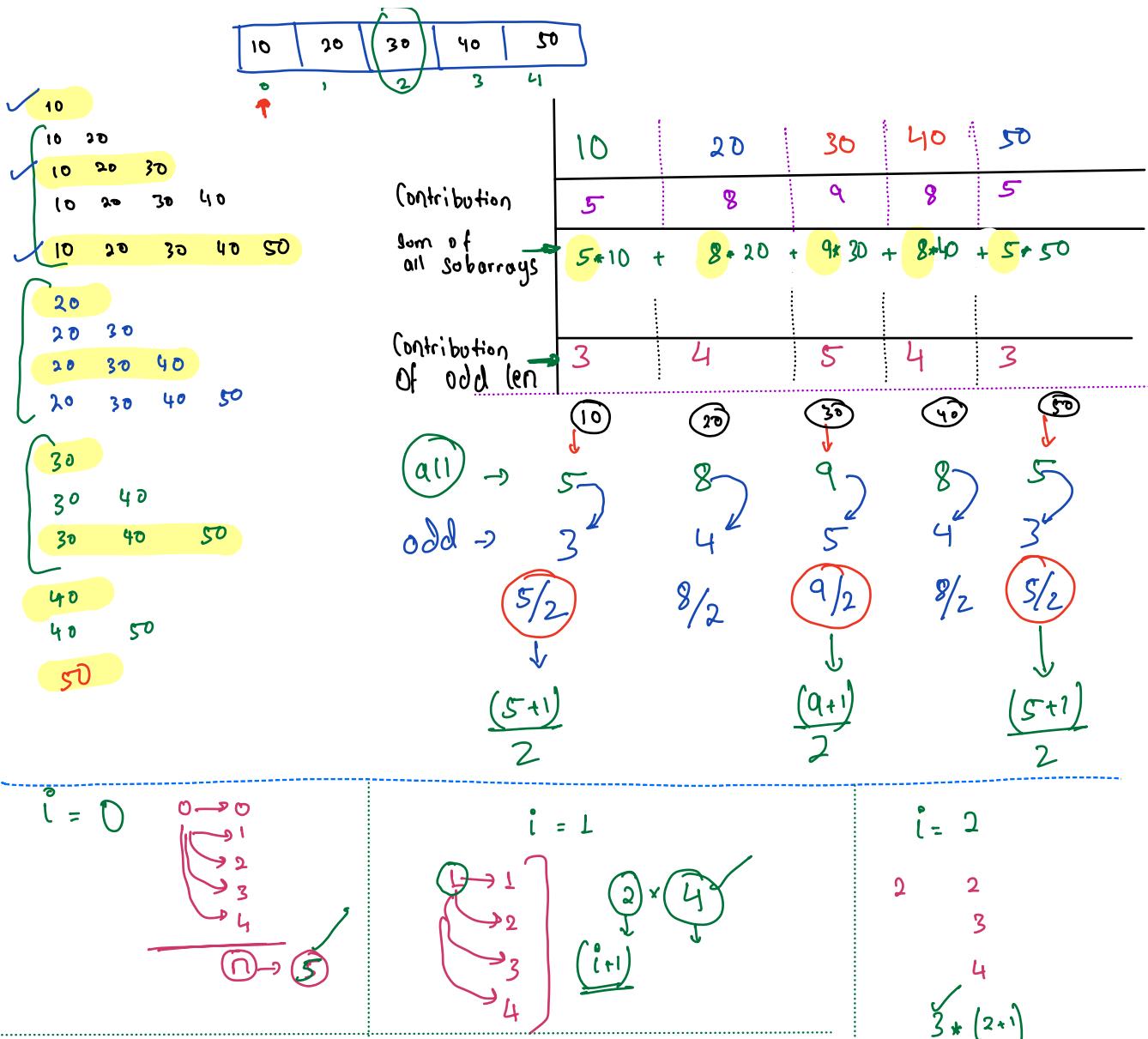
$$(10) + (10 + 20 + 30) + (20) + (20 + 30 + 40)$$
$$30 + 40$$

$$B.F = O(n^2)$$



$$(2 * 10) + 3 * 20 + 3 * 30 + 2 * 40$$

1 2 3 2



```
public class Solution {
    public int solve(int[] A) {
        int res = 0;
        int n = A.length;
        for(int i = 0 ; i < n; i++) {
            int contribution = (n-i)*(i+1);
            int countOfOdd = (contribution + 1)/2;
            res = res + (countOfOdd * A[i]);
        }
        return res;
    }
}
```

Sum of odd len.

$$\frac{(n-i)(i+1)+1}{2}$$

2. Little Pony and Maximum Element

Problem Description

Little Ponny is given an array, **A**, of **N** integers. In a particular operation, he can set any element of the array equal to **-1**.

He wants your help in finding out the minimum number of operations required such that the maximum element of the resulting array is **B**. If it is not possible, then return **-1**.

Problem Constraints

$1 \leq |A| \leq 10^5$

$1 \leq A[i] \leq 10^9$

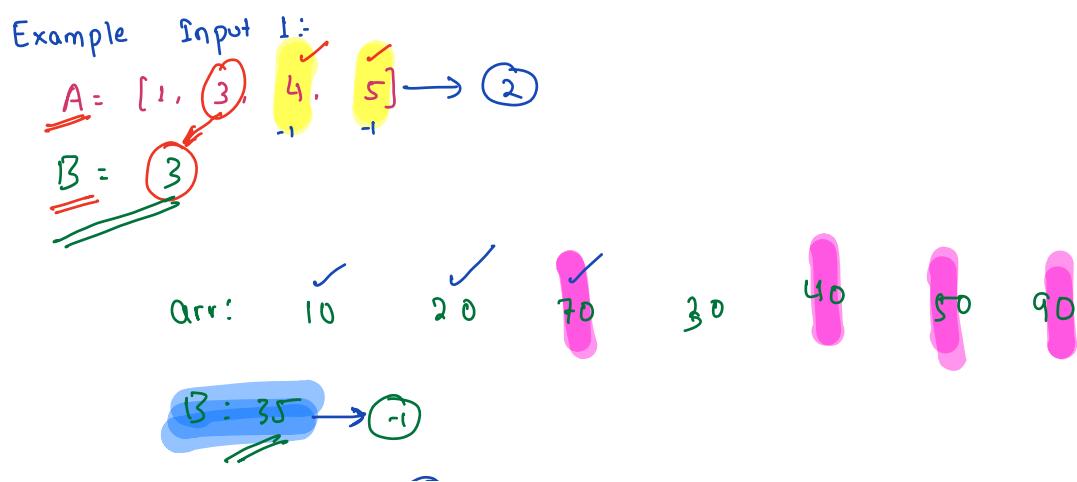
Input Format

The first argument of input contains an integer array, **A**.

The second argument of input contains an integer, **B**.

Output Format

Return an integer representing the answer.



$B = 30 \rightarrow 4$

```
public class Solution {  
    public int solve(int[] A, int B) {  
        int n = A.length, flag = 0, count = 0;  
        for(int i = 0 ; i < n; i++) {  
            if(A[i] == B) {  
                flag = 1;  
                break;  
            }  
        }  
        if(flag == 0) {  
            return -1;  
        }  
        for(int i = 0 ; i < n ; i++) {  
            if(A[i] > B) {  
                count++;  
            }  
        }  
        return count;  
    }  
}
```

3. Vowels in a Range.

Problem Description

Given a string **A** of length **N** consisting of lowercase letters, and **Q** queries given by the 2D array **B** of size **Q*2**. Each query consists of two integers **B[i][0]** and **B[i][1]**.

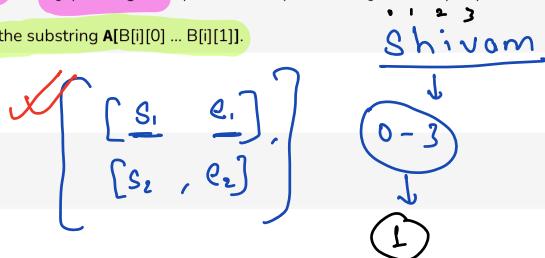
For every query, the task is to find the count of vowels in the substring **A[B[i][0]] ... B[i][1]]**.

Problem Constraints

$1 \leq N \leq 10^5$

$1 \leq Q \leq 10^5$

$0 \leq B[i][0] \leq B[i][1] < N$



Input Format

First argument **A** is a string.

Second argument **B** is a 2D array of integers.

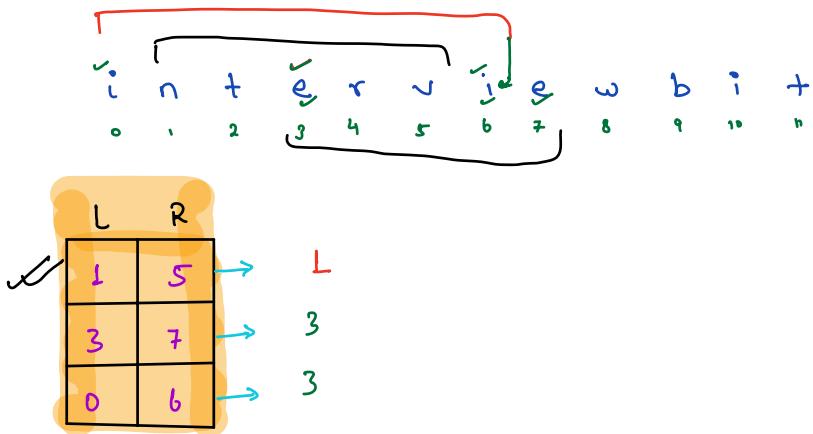
m o n a l i

Output Format

Return an array of integers.

0 - 5
3

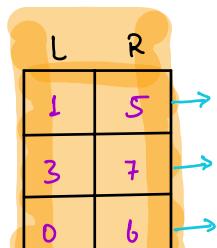
Range
Queries



Idea:-

Count of vowels in the array.
Using some generic formula of P.C

i n t e r n a t i o n a l
0 1 2 3 4 5 6 7 8 9 10 11



✓

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

→ pfCount

vowel count [2, 8] → 3

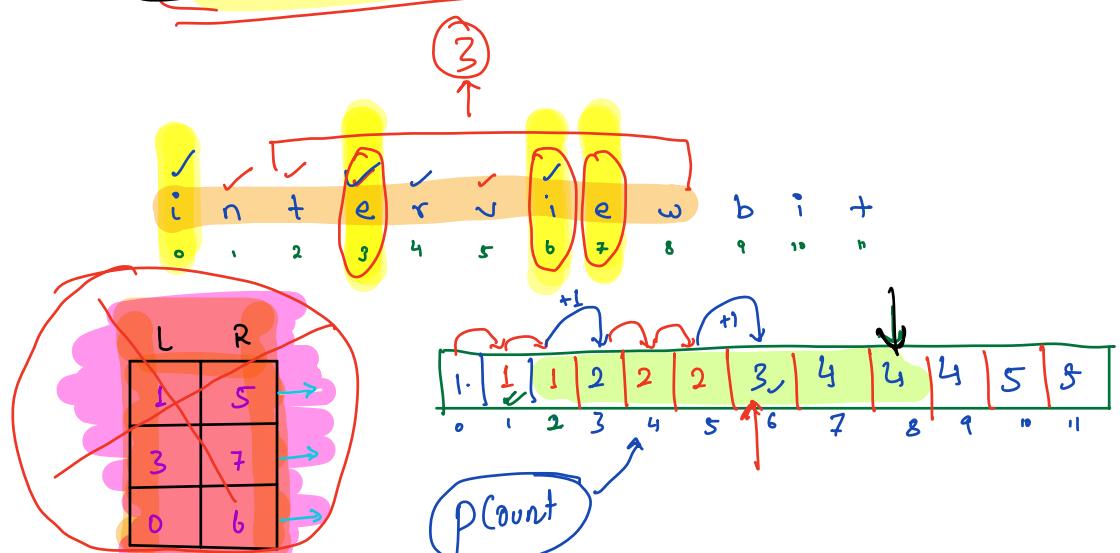
vowel count [3, 7] → 3

$$pfCount(A, B) = pfCount(B) - pfCount(A-1)$$

for $A > 0$

if $(A == 0)$

$$pfCount(A, B) = pfCount(B)$$

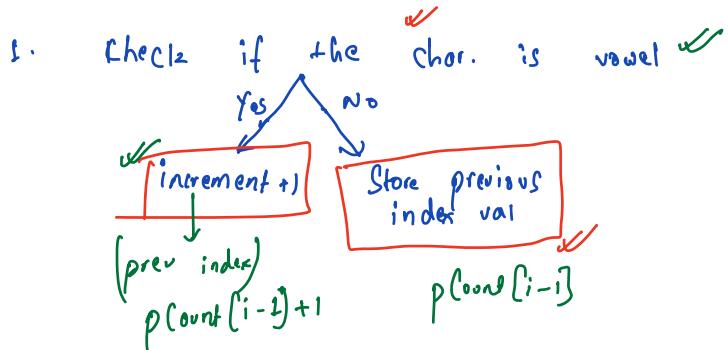


Queries

B[C][C]

$pfCount(6) - pfCount(0) \rightarrow 3 - 1$

$pfCount(8) \rightarrow 4$



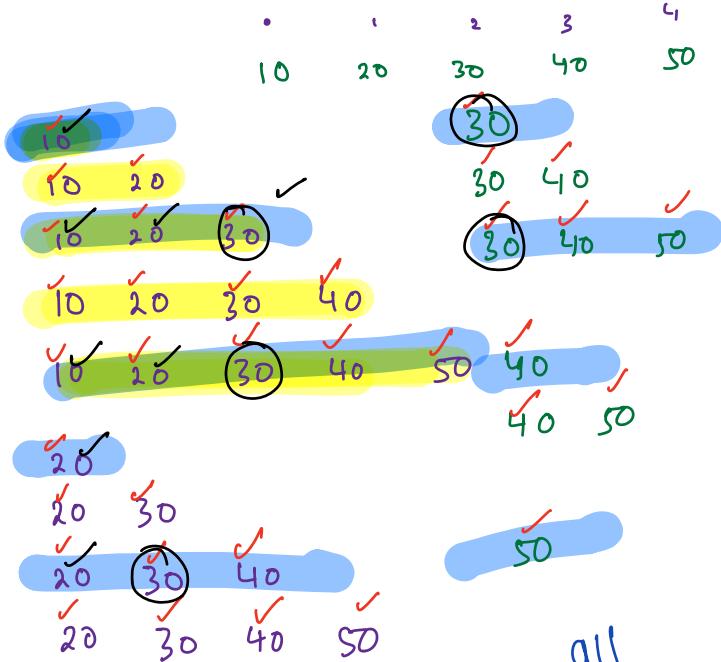
```

public class Solution {
    public int[] solve(String A, int[][] B) {
        int n = A.length();
        int res[] = new int[B.length];
        int pCount[] = new int[n];

        //Iterate and check if the character is vowels.
        // Find prefixCount
        if(isVowel(A.charAt(0))) {
            pCount[0] = 1;
        }
        for(int i = 1 ; i < n; i++) {
            char ch = A.charAt(i);
            if(isVowel(ch)) {
                pCount[i] = pCount[i-1] + 1;
            }
            else {
                pCount[i] = pCount[i-1];
            }
        }
        /* Loop over Range Query */
        for(int i = 0 ; i < B.length; i++) {
            int st = B[i][0];
            int end = B[i][1];
            if(st == 0) {
                res[i] = pCount[end];
            }
            else {
                res[i] = pCount[end] - pCount[st-1];
            }
        }
        return res;
    }

    public static boolean isVowel(char val) {
        if(val == 'a' || val == 'e' || val == 'i' || val == 'o' || val == 'u') {
            return true;
        }
        return false;
    }
}
  
```

Doubt Starts here



$$= \frac{n(n+1)}{2}$$

$$\frac{5 \times 6}{2} = 15$$

$$(10) + (10+20+30) + (10+20+30+40)$$

$$- - - - -$$

$$\begin{bmatrix} 20 & 30 & 40 & 50 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 9 & 8 & 7 & 6 \end{bmatrix}$$

all Subarray sum

odd sum

3	4	5	4	3
---	---	---	---	---

$$\left(\frac{(i+1)(n-i)+1}{2} \right)$$

$$\left(3*10 + 4*20 + 5*30 + \right)$$

$$\left(4*40 + 3*50 \right)$$

