

## Question 1

<https://www.geeksforgeeks.org/convert-a-number-from-base-2-to-base-6/#:~:text=Approach%3A%20The%20given%20problem%20can,to%20store%20the%20decimal%20number>

## Question 2

You are given a binary string (containing character 0 and 1 only)  $s$  of length  $n$ . Your task is to convert all the 1s in this string to 0. To do this you can do the following :

First choose an integer  $K$  and then perform the following operation any number of times :

Choose any substring of this string such that length of substring is greater than or equal to  $K$  and change all 0s to 1 and all 1s to 0 in that substring i.e. for a substring  $[L,R]$  ( $R-L+1 \geq K$ ) in  $s$ , for each  $i$  such that  $L \leq i \leq R$ , if  $s_i$  is 0 make it 1 and if it is 1 make it 0.

Find the maximum integer  $K$  less than or equal to  $n$  that we can choose such that it is possible to convert the string to all 0s using above method.

### Constraints

$1 \leq n \leq 100000$

### Examples

Input : 101

Output : 2

Explanation : If we choose  $K = 2$ , we can convert the string to all 0s by following way choose substring  $[0,1] = "10"$ , then  $s$  becomes 011

now choose substring  $[1,2] = "11"$ , then  $s$  becomes 000

It is the maximum K as we can not convert the string to all 0s by choosing  $K = 3$ .

Input : 10000

Output : 4

Explanation : We can choose  $K = 4$  and convert the following way

Choose substring  $[0,4] = "10000"$ , the s becomes 01111

Choose substring  $[1,4] = "1111"$ , then s becomes 00000

Input/Output

[execution time limit] 4 seconds (py3)

[input] string s

given binary string

[output] integer

Output the maximum K such that it is possible to convert s to all 0s by the operation.

GFG -

<https://www.geeksforgeeks.org/maximum-length-of-a-substring-required-to-be-flipped-repeatedly-to-make-all-characters-of-binary-string-equal-to-0/>

### Question 3

You are given n balloons(n is even), indexed from 0 to  $n - 1$ . Each balloon is painted with a number on it represented by an array nums. You are asked to burst all the balloons.

If you burst the two adjacent(let be  $i, i+1$ ) balloon, you will get  $\text{nums}[i - 1] * \text{nums}[i] * \text{nums}[i + 1] * \text{nums}[i + 2]$  coins. If  $i - 1$  or  $i + 2$  goes out of bounds of the array, then treat it as if there is a balloon with a 1 painted on it. After bursting  $i$  and  $i+1$ ,  $i-1$  and  $i+2$  would become adjacent.

Return the maximum coins you can collect by bursting the balloons wisely.

**Constraints:** $1 < n \leq 200$  $1 \leq \text{nums}[i] \leq 1000$ **TestCase:** $n=4$  $\text{nums}=[3,1,5,8]$ 

Explanation remove element at index 1,2 i.e(1,5) will get score=  $3 \times 1 \times 5 \times 8 = 120$

array would become [3,8] removing this two would get us score=  $1 \times 3 \times 8 \times 1 = 24$

 $\text{ans}=144$ **Similar Problem:**

similar problem but with one ballon burst instead of two

**Solution :**

<https://codeforces.com/blog/entry/99152#comment-879470>