Ques 1: You are given a sorted array of lenght N, with one element occuring more than N/2 times. Find that number.  
input: A[1,2,2,2,2,2,3]  
Output: 2

Ans: approach 1: moore voting algorithm

Approach 2: count the frequencies

Approach 3: as the lement are sorted and given more than n/2 occurrence so O(1)

Is simply find the mid and return the mid element

Ques 2);

Ques 2: A sorted array of unique integers and a cloned array with one element less.  
Find that missing element.  
A:  [1,2,3,4,5,6]  
A': [1,2,3,4,5]  
Output: 6

Ans:

Using xor

Ans 2: using linear search approach

Ans #: using binary search method

int misingValue(vector<int>&A,vector<int>B){  
    int low=0;  
    int high=A.size()-1;  
    while(low<high){  
        int mid=low+(high-low)/2;  
        if(A[mid]==B[mid])  
            low=mid+1;  
        else  
            high=mid;  
     }  
     return low;  
}

ques 3) Ques 3: You are given a string of N lowercase letters of the English alphabet.  
An operation consists of choosing a group of Kconsecutive equal characters and removing them.  
If you perform operations as long as it's possible, what is the final string you get?  
Test Case:  
Input:  
String: aab  
K=2  
Output: b  
  
Test Case:  
Input:  
String: aaabbb  
K=3  
input aabbba  
k=3  
Output: "" (Empty string)

approach 1:

Approach 1: Multiple iterations to remove the consecutive characters. High time complexity: O(N^2)

Asked to optimize

Approach2:

: Used two stack to solve the question. But the time complexity was O(N^2)