```
#include<bits/stdc++.h>
using namespace std;
int main(){
    string s;
    cin>>s;
    int n=s.length();
    int i=0,j=n-1;
    while(i<j){
        if(isalpha(s[i])&& isalpha(s[j]))
        {
            swap(s[i],s[j]);
            i++;
            j--;
        }
        else if((isalpha(s[i])&& !(isalpha(s[j]))))
            j--;
        else if(!(isalpha(s[i])) && isalpha(s[j]))
        i++;
        else
        {
            i++;
            j--;
        }
        }
        cout<<s;
    }
//two pointer and string manipulation
#include<bits/stdc++.h>
using namespace std;
int main(){
    int n;
    cin>>n;
    int k;
    cin>>k;
    int maxi=INT_MIN;
    int sums=0;
    int a[n];
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    for(int i=0;i<k;i++){
        sums+=a[i];
    }
    maxi=max(sums,maxi);
    for(int i=k;i<n;i++){
        sums+=a[i]-a[i-k];
        maxi=max(sums,maxi);
    }
    cout<<maxi;
}
```

```
//sliding window and maximum sum

#include<bits/stdc++.h>
using namespace std;
int main(){
    int n;
    cin>>n;
    int k;
    cin>>k;
    int a[n];
 ///   [4,1,7,6,8,13,10,2]
 //Output:25
}
```

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n, k;
    cin >> n >> k;

    int a[n];
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }

    int count = 0;

    for (int i = 0; i < k; i++) {
        if (a[i] % 2 != 0) {
            count++;
        }
    }

    int maxi = count;
    for (int i = k; i < n; i++) {

        if (a[i - k] % 2 != 0) {
            count--;
        }

        if (a[i] % 2 != 0) {
            count++;
        }
        maxi = max(maxi, count);
    }

    cout << maxi << "\n";

    return 0;
}

#include<bits/stdc++.h>
using namespace std;

int main() {
    int n, k;
    cin >> n >> k;

    int a[n];
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }

    int sums = 0, maxi = 0;
    int left = 0;

    for (int right = 0; right < n; right++) {
        sums += a[right];
        while (sums >= k) {
            maxi = max(maxi, right - left + 1);
            sums -= a[left];
            left++;
        }
    }

    cout << maxi << "\n";
    return 0;
}
```