```cpp
// check whether string is anagram or not
#include<iostream>
#include<vector>
#include<algorithm>
using namespace std;
bool Anagram(string str1, string str2){
    if(str1.length()!=str2.length()){
        return false;
    }
    vector<int> v(26,0);
    for(int i=0; i<str1.length(); i++){
        v[str1[i]]++;
        v[str2[i]]--;
    }
    for(int i=0; i<26; i++){
        if(v[i]!=0){
            return false;
        }
    }
    return true;
}
int main(){
    string str1 = "anagram", str2 = "naagram";
    if(Anagram(str1,str2)){
        cout<<"yes";
        return 0;
    }
    cout<<"no";
    return 0;
}




//longest palindrome you can make out of the given string
int main(){
    string str = "aaababbcc"; //aabcacbaa
    vector<int> v(26,0);
    for(int i=0; i<str.size(); i++){
        v[str[i]-'a']++;
    }
    int longestPlaindromeLength = 0;
    int odd = 0;
    for(int i=0; i<26;i++){
        if(v[i]%2 == 0){ //present even no. of times
            longestPlaindromeLength += v[i];
        }
        else{
            longestPlaindromeLength += v[i]-1;
            odd = 1;
        }
    }
    cout<<"Longest Palindrome Length is: "<<(longestPlaindromeLength+odd);
```

```
    return 0;
}



//return the length of binary substring containing maximum number of 1's you can replace
atmost k 0's
//01001011001 => max length of substring containig 1's is 5
int main(){
    string str = "01001011001";
    int maxLen = 0;
    int zeroCount = 0;
    int k = 2;
    int start = 0, end = 0;
    for (int end = 0; end < str.size(); end++) {
        if (str[end] == '0') {
            zeroCount++;
        }

        while (zeroCount > k) {
            if (str[start] == '0') {
                zeroCount--;
            }
            start++;
        }

        maxLen = max(maxLen, end - start + 1);
        cout<<maxLen<<endl;
    }

    cout << "Maximum length of substring with at most " << k << " zero flips is: " <<
maxLen << endl;
    return 0;
}



//Add Strings num1="2603126" , num2="569" , ans="2603695"
int main(){
    string num1 = "2603126", num2 = "569";
    string ans = "";
    int carry = 0, index1 = num1.size()-1, index2 = num2.size()-1;
    while(index2>=0){
        int num = carry + num2[index2]-'0' + num1[index1]-'0';
        carry = num/10;
        ans += (num%10)+'0';
        index1--;
        index2--;
    }
    while(index1>=0){
        int num = carry + num1[index1]-'0';
        carry = num/10;
        ans += (num%10)+'0';
        index1--;
```

```
        }
        if(carry){
            ans += '1';
        }
        reverse(ans.begin(), ans.end());
        cout<<ans<<endl;
        return 0;
    }
```