

Strings Problems

Problem 1: Palindrome string

Method

Compare first and last character, then move inward.

C++ solution

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

int main() {
    string s;
    cin >> s;

    int l = 0, r = (int)s.size() - 1;
    while (l < r) {
        if (s[l] != s[r]) {
            cout << "NO";
            return 0;
        }
        l++;
        r--;
    }

    cout << "YES";
    return 0;
}
```

Problem 2: Count vowels and consonants

C++ solution

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

bool isVowel(char c) {
    c = tolower(c);
    return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';
}

int main() {
    string s;
```

```

getline(cin, s);

int vowels = 0, consonants = 0;
for (char c : s) {
    if (isalpha(c)) {
        if (isVowel(c)) vowels++;
        else consonants++;
    }
}

cout << vowels << " " << consonants;
return 0;
}

```

Problem 3: Reverse words in a sentence

Input:

I love coding

Output:

coding love I

C++ solution

```

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

int main() {
    string s;
    getline(cin, s);

    stringstream ss(s);
    vector<string> words;
    string word;

    while (ss >> word) {
        words.push_back(word);
    }

    for (int i = (int)words.size() - 1; i >= 0; i--) {
        cout << words[i];
        if (i != 0) cout << " ";
    }
}

```

```
    return 0;
}
```

Problem 4: First non-repeating character

C++ solution

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

int main() {
    string s;
    cin >> s;

    vector<int> freq(256, 0);
    for (char c : s) freq[(unsigned char)c]++;

    for (char c : s) {
        if (freq[(unsigned char)c] == 1) {
            cout << c;
            return 0;
        }
    }

    cout << "-1";
    return 0;
}
```

Problem 5: Anagram check

Two strings are anagrams if they contain the same characters with same frequency.

C++ solution

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

int main() {
    string a, b;
    cin >> a >> b;

    if (a.size() != b.size()) {
        cout << "NO";
        return 0;
    }

    vector<int> count(256, 0);
```

```

for (char c : a) count[(unsigned char)c]++;
for (char c : b) count[(unsigned char)c]--;

for (int x : count) {
    if (x != 0) {
        cout << "NO";
        return 0;
    }
}

cout << "YES";
return 0;
}

```

Problem 6: Remove duplicate characters

Preserve first occurrence order.

C++ solution

```

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

int main() {
    string s;
    cin >> s;

    vector<int> seen(256, 0);
    string result = "";

    for (char c : s) {
        if (!seen[(unsigned char)c]) {
            seen[(unsigned char)c] = 1;
            result += c;
        }
    }

    cout << result;
    return 0;
}

```