Assignment Solutions | Strings - 2 | Week 7

1. Input a string and concatenate with its reverse string and print it.

   Input : str = "PWSkills"

   Output : "PWSkillssllikSWP"

   Input : str = "pw"

   Output : "pwwp"

Solution :

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    string s;
    cin >> s;
    string temp = s;
    reverse(temp.begin(), temp.end());
    s += temp;
    cout << s << '\n';
}
```

2. Find the second largest digit in the string consisting of digits from ‘0’ to ‘9’.

   Input : str = "2947578"

   Output : 8

   Input : str = "1241"

   Output : 2

Solution :

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

int main() {
    string p;
    cin >> p;
    char f = '?', s = '?';
    for (auto &i : p) {
        if (f == '?' || i > f) {
            s = f;
            f = i;
        } else if (i > s) {
            s = i;
        }
    }
    cout << s;
}
```

3. Input a string and return the number of substrings that contain only vowels.

   Input : str = "abjkoe"

   Output : 4

   Explanation : The possible substrings that only contain vowels are "a" , "o" , "e" , "oe"

   Input : str = "hgdhpw"

   Output : 0

Solution :

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    string s;
    cin >> s;

    int ans = 0, l = 0;
    for (int i = 0; i < (int)s.size(); i++) {
        if (s[i] == 'a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'o' || s[i] == 'u')
l++;
        else {
            ans = l * (l + 1) / 2;
            l = 0;
        }
    }
    ans += l * (l + 1) / 2;
    cout << ans;
}
```

4. Given an array of strings. Check whether they are anagram or not.

   Input : s = "car" , t = "arc"

   Output : True

Input : s = "book" , t = "hook"

Output : False

Solution :

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

int main() {
    int n;
    cin >> n;
    vector<string> s(n);
    for (auto &i : s) cin >> i;

    bool ok = true;


    for (int i = 0; i < n; i++) {
        sort(s[i].begin(), s[i].end());
        ok &= (s[i] == s[0]);
    }

    cout << (ok ? "YES" : "NO");
}
```

5. Given a sentence ‘str’, return the word that is lexicographically maximum.

   Input : str = "proud to be pwians"

   Output : pwians

   Input : str = "decode dsa with pw"

   Output : with

Solution :

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    string str ;
     getline(cin , str); //method to input a string with spaces
    int n = str.size();

    string mx = "", word = "";

    for (int i = 0; i < n; i++) {
        if (str[i] == ' ') {
            mx = max(mx, word);
            word = "";
        } else {
            word += str[i];
        }
    }
    mx = max(mx, word);
    cout << mx << '\n';
}
```
