Kaarthik Raja CSE

DSA-2

ANAGRAM PROBLEM

```
// } Driver Code Ends
// User function Template for C++
class Solution {
  public:
    int areAnagram(string s1, string s2) {
        // code here
        sort(s1.begin(),s1.end());
        sort(s2.begin(),s2.end());
        if(s1==s2){
            return 1;
        }
        return 0;
    }
};
// } Driver Code Ends
```

MAX ROW WITH 1’S

```
// User function template for C++
class Solution {
  public:
    int rowWithMax1s(vector<vector<int> > &arr) {
        // code here
    int count = 0;
        int ans = 0;
        int index = -1;


        for(int i = 0; i < arr.size(); i++) {
            count = 0;
            for(int j = 0; j < arr[i].size(); j++) {
                if(arr[i][j] == 1) {
                    count++;
                }
            }

            if(count > ans) {
                ans = count;
                index = i;
            }
        }
        return index;
    }
};
// } Driver Code Ends
```

LONGEST CONSECUTIVE SUBSEQUENCE

```python
class Solution:

    # arr[] : the input array

    #Function to return length of longest subsequence of consecutive integers.
    def findLongestConseqSubseq(self,arr):
        #code here
        d={}
        l=[]
        arr=set(arr)
        for i in arr:
            if i-1 not in arr and d.get(i)!=True:
                c=0
                while i in arr:
                    d[i]=True
                    c=c+1
                    i=i+1
                l.append(c)
        return max(l)



# } Driver Code Ends
```

LONGEST PALINDROME IN A STRING

```python
def LongestPalindrome(s):
    def expand(l, r):
        while l >= 0 and r < len(s) and s[l] == s[r]:
            l -= 1
            r += 1
        return s[l + 1:r]

    result = ""
    for i in range(len(s)):
        odd = expand(i, i)
        even = expand(i, i + 1)

        if len(odd) > len(result):
            result = odd
        if len(even) > len(result):
            result = even

    return result



# } Driver Code Ends
```