CodeaCats Week 6

1.

```
char findTheDifference(string s, string t) {
    // sort(s.begin(),s.end());
    // sort(t.begin(),t.end());
    // int i;
    // for(i=0;i<s.length();i++)
    //     if(s[i]!=t[i])
    //         return t[i];
    // return t[i];

    char v=0;
    for( char c:s)
        v^=c;
    for( char c:t)
        v^=c;
    return v;
}
```

2.

```
bool check_duck(string N){
    //code
    if(N[0]=='0')
        return false;
    for(int i=0;i<N.size();i++)
    {
        if(N[i]=='0')
            return true;
    }
    return false;
}
```

3.

```
class Solution {
public:
    int longestSubstring(string s, int k) {

        int len=s.length();

        if(len==0 || len<k)
            return 0;
        if(k<=1)
```

```
        return len;

    unordered_map<char,int> mp;
    for(char c:s)
        mp[c]++;

    int left =0;

    while(left < len && mp[s[left]] >= k)
        left++;

    if(left >= len-1)
        return left;

    int L = longestSubstring(s.substr(0,left) ,k);

    while(left < len && mp[s[left]] < k)
        left++;

    int R = left < len ? longestSubstring(s.substr(left),k) : 0;


    return max(L,R);
  }
};
```

4.

```
 int romanToInt(string s) {
    unordered_map<char, int> mp = {{'M', 1000}, {'D', 500}, {'C', 100}, {'L', 50}, {'X', 10}, {'V', 5},
{'I', 1}};
        int res = mp[s.back()];
        for(int i = 0; i < s.size() - 1; i++) {
                if(mp[s[i]] < mp[s[i + 1]]) res -= mp[s[i]];
                else res += mp[s[i]];
        }
        return res;
  }
```

5.

```
int numJewelsInStones(string jewels, string stones) {
       int ctr=0;
        for(int i=0;i<jewels.length();i++){
            for(int j=0;j<stones.length();j++)
```

```
                if(jewels[i]==stones[j])
                    ctr++;
        }
        return ctr;
    }

    //using set
    int numJewelsInStones(string jewels, string stones) {
       int ctr=0;
        unordered_set<char> setJewels(jewels.begin(),jewels.end());
        for(char s:stones)
            if(setJewels.count(s))
                ctr++;
        return ctr;
    }
```

6.

```
string interpret(string command) {
        string res;
        int i=0;
        while(command[i])
        {
            if(command[i]=='G'){
                res += 'G';
                i+=1;
            }
            else if(command[i] == '(' && command[i+1] == ')'){
                res += 'o';
                i+=2;
            }
            else if(command[i] == '(' && command[i+3] == ')'){
                res += "al";
                i += 4;
            }
        }
        return res;
    }
```

7.

```
bool checkAlmostEquivalent(string word1, string word2) {
        int alpha[26]={};
        for( char c: word1)
            alpha[c-'a']++;
        for( char c: word2)
            alpha[c-'a']--;
        for(int i=0;i<26;i++)
            if(abs(alpha[i])>3)
                return false;
       return true;
    }
```