WORKSHEET 3

Student Name: Vivek Kumar
UID: 21BCS8129
DOMAIN CAMP: 16-01-2023 to 28-01-2023
Section/Group: DWWC-77
Subject Name: IT Skills (DSA)

Question 1. CHEF AND LOCKOUT DRAWS

Language: C++14

```cpp
#include <iostream>
using namespace std;

int main() {
    int T;
    int A,B,C;
    cin>>T;
    for(int i=0;i<T;i++)
    {
        cin>>A>>B>>C;
        if(A==(B+C))
        {
            cout<<"YES"<<endl;
        }
        else if(B==(A+C))
        {
            cout<<"YES"<<endl;
        }
        else if(C==(A+B))
        {
            cout<<"YES"<<endl;
        }
        else
        {
            cout<<"NO"<<endl;
        }
    }
    // your code goes here
    return 0;
}
```

0:0

SOLUTION:

Question 2.

Status: ✔ Correct Answer — Submission ID: 84476442

Score: 100 | Time: 0.01s | Memory: 5.2M

Sub-Task	Task #	Result (time)
1	0	AC (0.003468)
1	1	AC (0.006011)
1	2	AC (0.005706)
Subtask Score: 100.00%		Result - AC
		Total Score = 100.00%

CHEF AND PATIENTS

Language: C++14

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

void solve() {
    int n ; cin >> n;
    vector<pair<int, int>> a(n);
    int mx = 0;
    for(int i=0; i<n; i++) {
        a[i].second = i;
        cin >> a[i].first;
        mx = max(mx, a[i].first);
    }
    for(int i=0; i<n; i++) a[i].first = mx - a[i].first;

    sort(a.begin(), a.end());
    vector<int> ans(n);

    for(int i=0; i<n; i++) ans[a[i].second] = i+1;

    for(int& e : ans) cout << e << " "; cout << endl;
}

int main() {
    int t; cin >> t;
    while(t--) solve();
    return 0;
}
```

SOLUTION:

Question 3.

Status: ✔ Correct Answer — Submission ID: 84479650

Score:	Time:	Memory:
100	0.15s	5.1M

Sub-Task	Task #	Result (time)
1	0	AC (0.149032)
Subtask Score: 100.00%		Result - AC
		Total Score = 100.00%

Question 4.

CANDIES

Language: C++14

```cpp
#include <iostream>
#include <climits>
using namespace std;

int main() {
    // your code goes here
    int t;cin>>t;
    while(t--){
        int n;cin>>n;
        int arr[n];
        for(int i=0;i<n;i++) cin>>arr[i];
        int a=INT_MIN , b=INT_MAX;
        for(int i=0;i<n;i++){
            if(arr[i]<b) b=arr[i];
            if(arr[i]>a) a=arr[i];
        }
        int diff = a-b;
        int count = 0;
        for(int i=0;i<n;i++){
            if(diff==arr[i]) count++;
        }
        if(count>0) cout<<diff<<endl;
        else cout<<"NO"<<endl;

    }
    return 0;
}
```

0:0

SOLUTION:

Status: ✔ Correct Answer Submission ID: 84476061

Time:	Memory:
0.00s	5.3M

Question 5.

CHEF AND PARTY

Language: C++14

```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int t;
    cin >> t;
    while (t--)
    {
        int n;
        cin >> n;
        int *arr = new int[n];
        for (int i = 0; i < n; i++)
        {
            cin >> arr[i];
        }
        sort(arr, arr + n);
        int ans = 0;
        for (int i = 0; i < n; i++)
        {
            if (arr[i] <= ans)
            {
                ans++;
            }
        }
        cout << ans << endl;
    }
    return 0;
}

```

0:0

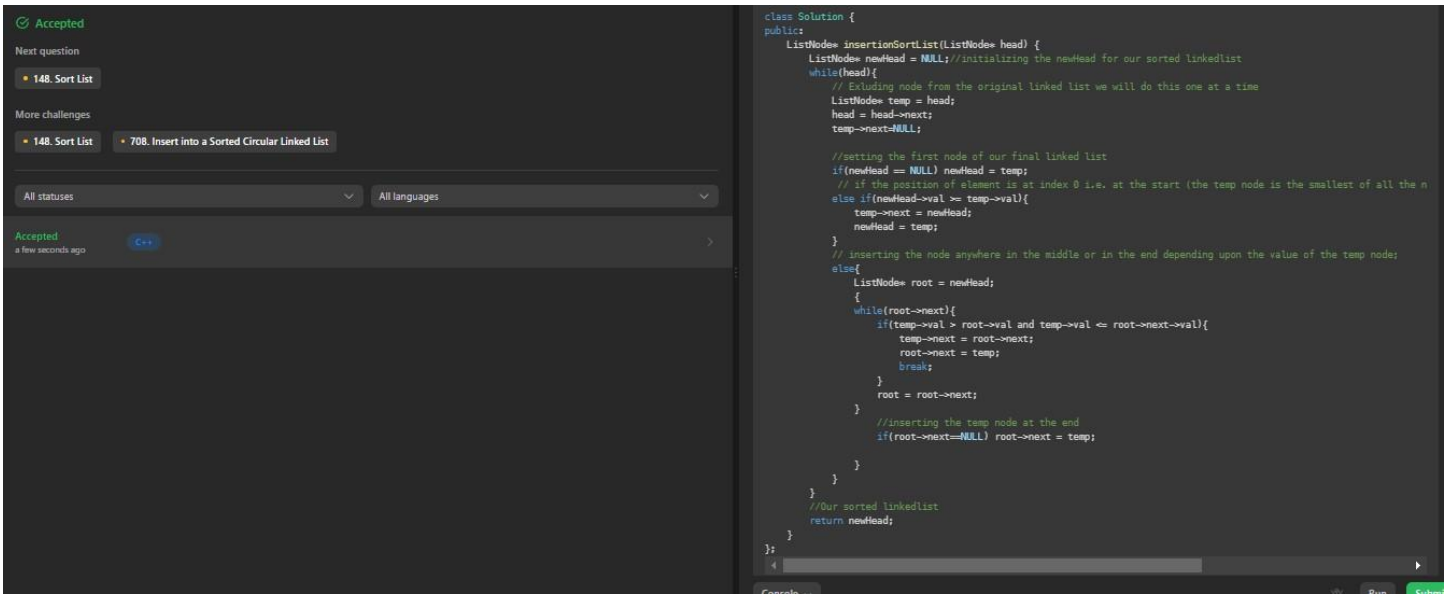
SOLUTION:

Status: ✔ Correct Answer Submission ID: 84847118

Time: 0.18s Memory: 7.4M

Question 6.

INSERTION SORT LIST

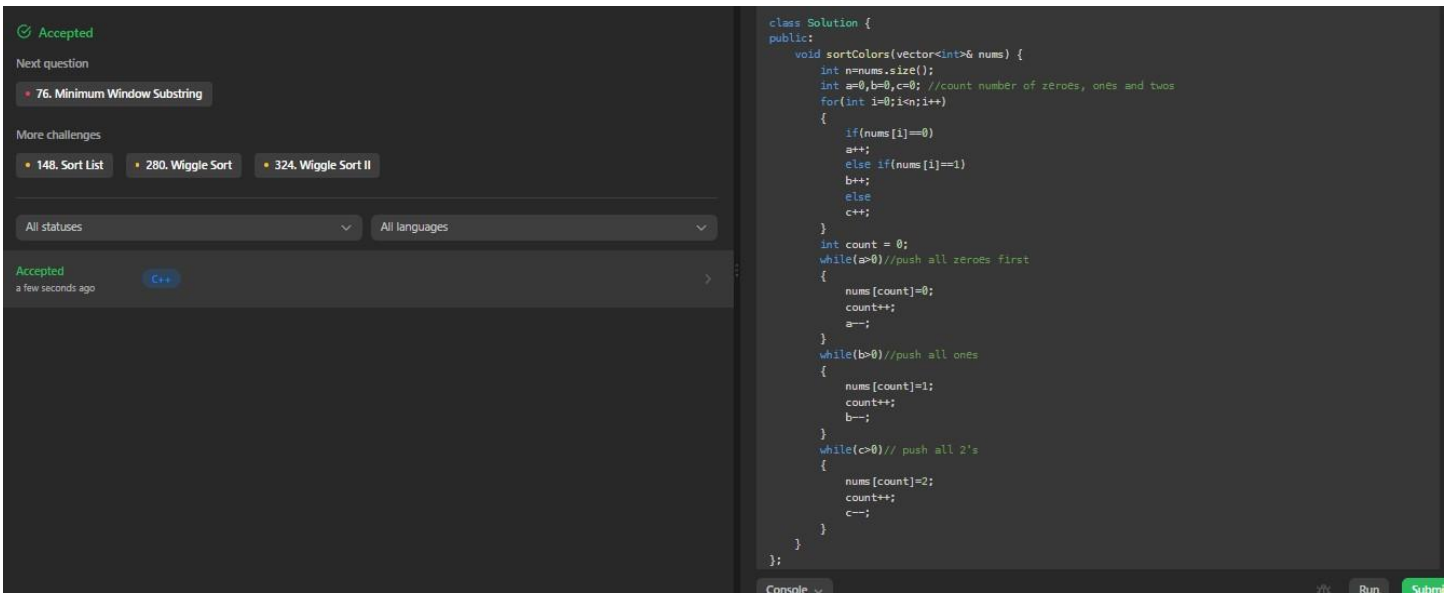



```
class Solution {
public:
    ListNode* insertionSortList(ListNode* head) {
        ListNode* newHead = NULL;//initializing the newHead for our sorted linkedlist
        while(head){
            // Exluding node from the original linked list we will do this one at a time
            ListNode* temp = head;
            head = head->next;
            temp->next=NULL;

            //setting the first node of our final linked list
            if(newHead == NULL) newHead = temp;
             // if the position of element is at index 0 i.e. at the start (the temp node is the smallest of all the n
            else if(newHead->val >= temp->val){
                temp->next = newHead;
                newHead = temp;
            }
            // inserting the node anywhere in the middle or in the end depending upon the value of the temp node;
            else{
                ListNode* root = newHead;
                {
                while(root->next){
                    if(temp->val > root->val and temp->val <= root->next->val){
                        temp->next = root->next;
                        root->next = temp;
                        break;
                    }
                    root = root->next;
                }
                    //inserting the temp node at the end
                    if(root->next==NULL) root->next = temp;

                }
            }
        }
        //Our sorted linkedlist
        return newHead;
    }
};
```

Question 6. COLOR SORT




```
class Solution {
public:
    void sortColors(vector<int>& nums) {
        int n=nums.size();
        int a=0,b=0,c=0; //count number of zeroes, ones and twos
        for(int i=0;i<n;i++)
        {
            if(nums[i]==0)
            a++;
            else if(nums[i]==1)
            b++;
            else
            c++;
        }
        int count = 0;
        while(a>0)//push all zeroes first
        {
            nums[count]=0;
            count++;
            a--;
        }
        while(b>0)//push all ones
        {
            nums[count]=1;
            count++;
            b--;
        }
        while(c>0)// push all 2's
        {
            nums[count]=2;
            count++;
            c--;
        }
    }
};
```