

WORKSHEET 4

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. TEMPLE LAND (ARRAY)

```
Language: C++14

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      // your code goes here
6      int t;
7      cin>>t;
8      while(t--){
9          int n;
10         cin>>n;
11         int arr[n];
12         for(int i=0;i<n;i++){
13             cin>>arr[i];
14         }
15         bool flag=true;
16
17         if((n%2==0) || arr[0]!=1) flag=false;
18         // else if(arr[n/2]>arr[n/2+1] || arr[n/2]<=arr[n/2-1]) flag=false;
19         else{
20             int start=0, end=n-1, count=1;
21             while(start<end){
22                 if(arr[start]!=arr[end] || arr[start]!=count){
23                     flag=false;
24                     break;
25                 }
26                 start++,end--;
27                 count++;
28             }
29         }
30         if(flag) cout<<"yes"<<endl;
31         else
32             cout<<"no"<<endl;
33     }
34     return 0;
35 }
```

SOLUTION:

Status:  Correct Answer Submission ID: [84576000](#)


Time: 0.01s	Memory: 5.3M
----------------	-----------------

Question 2. ROADS IN CHEFLAND (ARRAY)

```
Language: C++14

1  #include <iostream>
2  using namespace std;
3  int main() {
4      long long t,n,cost,v;
5      cin>>t;
6      while(t--){
7          cin>>n;
8          cost=0;
9          if(!(n&(n-1))){
10             cout<<"-1\n";
11             continue;
12          }
13          for(int i=1;i<=n;i<=1){
14             v=(n-i)/(i<1);
15             cost+=v*i +i;
16          }
17          cost--;
18          cout<<cost<<"\n";
19      }
20      return 0;
21 }
```

SOLUTION:

Status:  Correct Answer Submission ID: [84576100](#)

Time: 0.38s	Memory: 5.2M
----------------	-----------------

Question 3. SOLVE THE CASE (ARRAY)

```
Language: C++14

1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5
6 int main(){
7     int t;cin>>t;
8     for(int j=0;j<t;j++){
9         int n;cin>>n;
10        vector<int> v1,v2;
11        for(int i=0;i<n;i++){
12            int a; cin>>a;
13            v1.push_back(a);
14        }
15        for(int i=0;i<n;i++){
16            int temp=v1[i];
17            v2.push_back(temp);
18            while(temp==v1[i] && i<n){
19                i++;
20            }
21            i--;
22        }
23        int k=(int)v2.size();
24        for(int i=0;i<k;i++){
25            cout<<v2[i]<<" ";
26        }
27        cout<<"\n";
28    }
29    return 0;
30 }
31
```

SOLUTION:

Status: ✓ Correct Answer

Submission ID: [85014028](#)

Score: 100

Time: 0.00s

Memory: 5.3M

Sub-Task	Task #	Result (time)
1	0	AC (0.003930)
Subtask Score: 30.00%		Result - AC
2	1	AC (0.003999)
Subtask Score: 30.00%		Result - AC
3	2	AC (0.003919)
Subtask Score: 40.00%		Result - AC
Total Score = 100.00%		

Question 4. COOKING MACHINE (ARRAY)

```

Language: C++14

1  #include <bits/stdc++.h>
2  using namespace std;
3  bool is(int n){
4      while(n>0){
5
6          int c=n%2;
7          if(c!=0 && n!=1){
8              return false;
9          }
10         n=n/2;
11     }
12     return true;
13 }
14
15 int main() {
16     int t;
17     cin>>t;
18     while(t--){
19         int a,b;
20         cin>>a>>b;
21         int counter=0;
22         while(!is(a)){
23             if(a==1) break;
24             if(a%2!=0){ a=(a-1)/2;
25                 counter++;
26             }
27             else {a=a/2;
28                 counter++;}
29         }
30     }
31
32     if(b>a){
33         while(b>a){
34
35             a=a*2;
36             counter++;
37             if(b==a) break;
38         }
39     }else if(b<a){
40         while(a>b){
41
42             a=a/2;
43             counter++;
44             if(b==a) break;
45         }
46     }
47     cout<<counter<<endl;
48
49 }
50
51 return 0;
52 }

```

SOLUTION:

Status: ✔ Correct Answer

Submission ID: [85014150](#)

Score: 100

Time: 0.00s

Memory: 5.3M

Sub-Task	Task #	Result (time)
1	0	AC (0.003955)
1	1	AC (0.004388)
1	2	AC (0.004055)
1	3	AC (0.004158)
Subtask Score: 40.00%		Result - AC
2	4	AC (0.004246)
2	5	AC (0.004200)
2	6	AC (0.004046)
2	7	AC (0.004269)
Subtask Score: 60.00%		Result - AC
Total Score = 100.00%		

Question .

5 GOLD COLLECTION (ARRAY)

```
Language: C++14
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int t;
6      scanf("%d",&t);
7      while(t-->0)
8      {
9          int n;
10         scanf("%d",&n);
11
12         int arr[n];
13         for(int i=0;i<n;i++)
14         {
15             scanf("%d",&arr[i]);
16         }
17         for(int i=1;i<n;i++)
18         {
19             arr[i]+=arr[i-1];
20         }
21         int q;
22         scanf("%d",&q);
23         for(int i=0;i<q;i++)
24         {
25             int q1,q2;
26             scanf("%d%d",&q1,&q2);
27             if(q1==1)
28             {
29                 printf ("%d\n", arr[q2-1]);
30             }
31             else
32             {
33                 printf("%d\n", arr[q2-1]-arr[q1-2]);
34             }
35         }
36     }
37     return 0;
38 }
```

SOLUTION:

Status: ✓ Correct Answer Submission ID: 85014324

Time:	Memory:
0.19s	5.3M

Question .

Question 6. PALINDROME LINKED LIST (LINKED LIST)

Accepted

Next question

235. Lowest Common Ancestor of a Binary Search Tree

More challenges

9. Palindrome Number 125. Valid Palindrome 206. Reverse Linked List

All statuses All languages

Accepted
a few seconds ago

C++

```

/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    bool isPalindrome(ListNode* head) {
        vector<int>v;
        while(head){
            v.push_back(head->val);
            head=head->next;
        }
        for(int i=0; i<v.size()/2; i++){if(v[i]!=v[v.size()-1-i]){return false;}}
        return true;
    }
};

```

7 ADD TWO NUMBERS (LINKED LIST)

Accepted

Next question

3. Longest Substring Without Repeating Characters

More challenges

43. Multiply Strings 67. Add Binary 371. Sum of Two Integers

All statuses All languages

Accepted
a few seconds ago

C++

```

class Solution {
public:
    ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
        ListNode* dummy=new ListNode();
        ListNode* temp=dummy;
        int carry=0;
        while(l1!=NULL || l2!=NULL || carry){
            int sum=0;
            if(l1!=NULL){
                sum+=l1->val;
                l1=l1->next;
            }
            if(l2!=NULL){
                sum+=l2->val;
                l2=l2->next;
            }
            sum+=carry;
            carry=sum/10;
            ListNode* newnode=new ListNode(sum%10);
            temp->next=newnode;
            temp=temp->next;
        }
        return dummy->next;
    }
};

```


Question .

Question 8. MIDDLE OF THE LINKED LIST (LINKED LIST)

Accepted

Next question

877. Stone Game

More challenges

2095. Delete the Middle Node of a Linked List

2130. Maximum Twin Sum of a Linked List

All statuses

All languages

Accepted

a few seconds ago

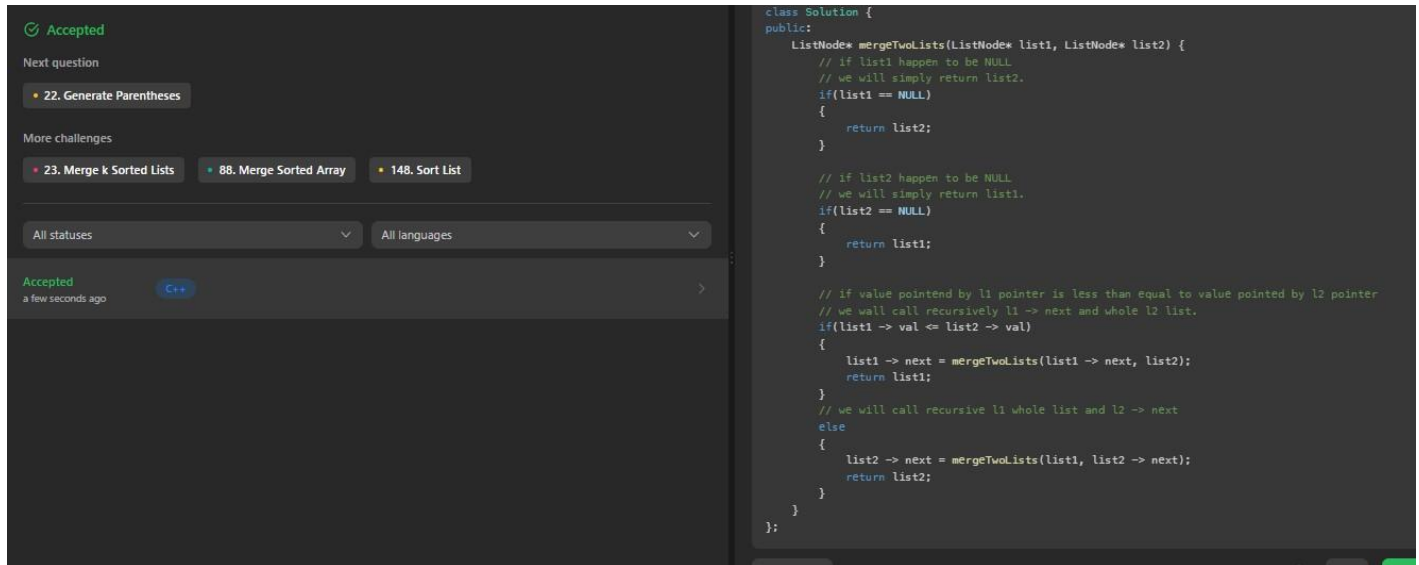
C++

```
class Solution {
public:
    ListNode* middleNode(ListNode* head) {
        //Naive Approach->
        //if there is only single Node
        if(head->next==NULL)
            return head;
        //Find length of Linked List Using Loop
        //Store Head in another ListNode pointer for traversing and count=0

        ListNode *curr=head;
        int c=0;
        while(curr!=NULL)
        {
            curr=curr->next;
            c++;
        }
        //Now Store Head in other Pointer Variable
        curr=head;
        //Traverse curr c/2 times and u get middle
        for(int i=0;i<c/2;i++)
        {
            curr=curr->next;
        }
        return curr;
    }
};
```


Question .

9 MERGE TWO SORTED LISTS (LINKED LIST)



The screenshot displays a coding interface with a sidebar on the left and a main editor on the right. The sidebar shows a list of challenges, with '23. Merge k Sorted Lists' selected. Below the list, there are filters for 'All statuses' and 'All languages'. The main editor shows a C++ solution for the 'Merge Two Sorted Lists' problem. The solution is marked as 'Accepted' and was submitted 'a few seconds ago'. The code is as follows:

```
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
        // if list1 happen to be NULL
        // we will simply return list2.
        if(list1 == NULL)
        {
            return list2;
        }

        // if list2 happen to be NULL
        // we will simply return list1.
        if(list2 == NULL)
        {
            return list1;
        }

        // if value pointed by l1 pointer is less than equal to value pointed by l2 pointer
        // we will call recursively l1 -> next and whole l2 list.
        if(list1 -> val <= list2 -> val)
        {
            list1 -> next = mergeTwoLists(list1 -> next, list2);
            return list1;
        }
        // we will call recursive l1 whole list and l2 -> next
        else
        {
            list2 -> next = mergeTwoLists(list1, list2 -> next);
            return list2;
        }
    }
};
```

Question .

Question 10. SPLIT LINKED LIST IN PARTS (LINKED LIST)

Accepted

Next question

726. Number of Atoms

More challenges

61. Rotate List 328. Odd Even Linked List

All statuses All languages

Accepted
a few seconds ago C++

```

class Solution {
public:
    vector<ListNode*> splitListToParts(ListNode* head, int k) {
        int len=0;
        ListNode* temp=head;
        while(temp)
        {
            len++;
            temp=temp->next;
        }

        int numNodes=len/k; //the number of nodes that are to be in each group
        int extra=len%k; //the extra nodes that are left
        int i=0;
        vector<ListNode*> res;
        temp=head;
        while(temp)
        {
            res.push_back(temp);
            //get the numNodes and make the last node next to NULL
            int curLen=1;
            while(curLen<numNodes)
            {
                temp=temp->next;
                curLen++;
            }
            temp=temp->next;
            curLen++;
        }
        if(extra>0 && len>k)
        {
            temp=temp->next;
            extra--;
        }
        ListNode* x=temp->next;
        temp->next=NULL;
        temp=x;
    }
    //if the number of nodes are less than k we add NULL
    while(len<k)
    {
        res.push_back(NULL);
        len++;
    }
    return res;
};

```