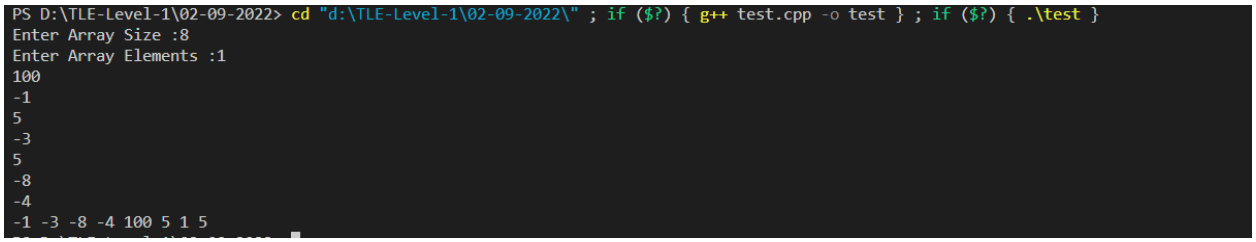
Name : Dev Adnani
SID : 202212012
Subject : DSA
Topic : Array-LinkedList
Lab : 03

Q1 :

```
#include <bits/stdc++.h>
using namespace std;
#define endl '\n'

void rotate(int arr[], int d, int d2)
{
	int temp = arr[d];
	for (int j = d; j > d2 - 1; j--)
	{
	arr[j] = arr[j - 1];
	}
	arr[d] = temp;
}

void change(int arr[], int n)
{
	int ll= -1;

	for (int i = 0; i < n; i++)
	{
	if (arr[i] < 0)
	{
	ll+= 1;
	int temp = arr[i];
	arr[i] = arr[ll];
	arr[ll] = temp;
	if (i - ll>= 2)
	rotate(arr, ll+ 1, i);
	}
	}
}

int main()
{
	int x;
	cout << "Enter Array Size :";
	cin >> x;

	int arrx[x];

	cout << "Enter Array Elements :";
	for (int i = 0; i < x; i++)
```

```
        cin >> arrx[i];

        change(arrx, x);

        for (int i = 0; i <x; i++)
        {
        cout << arrx[i] << " ";
        }
}
```

O/P :



Q2:

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

int main()
{
	int x;
	cout << "Enter Array Size :";
	cin >> x;

	int arrx[x];

	cout << "Enter Array Elements :";
	for (int i = 0; i < x; i++)
	{
	cin >> arrx[i];
	}
	for (int i = 0; i < x; i++)
	{
	int cn = 0;
	for (int j = 0; j < x; j++)
	{
	if (arrx[i] == arrx[j])
	{
		cn++;
	}
	}

	if (cn >= x / 2)
	{
	cout <<"This Element Occurs N/2 Times: "<< arrx[i];
	break;
	}
	}
}
```

O/P :

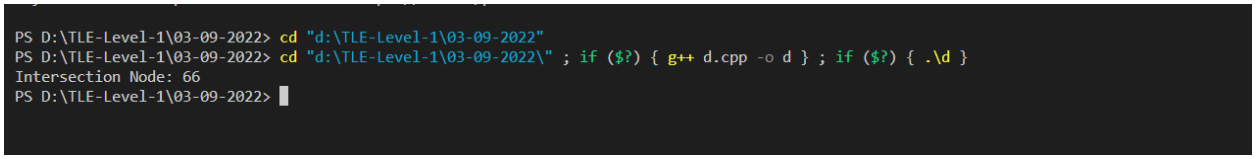
```
PS F:\DSALAB#> cd "f:\DSALAB#"
PS F:\DSALAB#> cd "f:\DSALAB#\" ; if ($?) { g++ test.cpp -o test } ; if ($?) { .\test }
Enter Array Size :7
Enter Array Elements :1
1
2
2
1
1
4
This Element Occurs N/2 Times: 1
PS F:\DSALAB#>
```

Q3 :

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

struct Node {
   int data;
   Node* next;
};

Node* intersectionNode(Node* linkListOne, Node* linkListTwo)
{
   while (linkListTwo) {
          Node* temp = linkListOne;
          while (temp) {
                 if (temp == linkListTwo)
                        return linkListTwo;
                 temp = temp->next;
          }
          linkListTwo = linkListTwo->next;
   }
   return NULL;
}

int main()
{
   Node* nextNode;

   Node* linkListOne = new Node();
   linkListOne->data = 1;

   Node* linkListTwo = new Node();
   linkListTwo->data = 2;

   nextNode = new Node();
   nextNode->data = 3;
   linkListTwo->next = nextNode;

   nextNode = new Node();
   nextNode->data = 4;
   linkListTwo->next->next = nextNode;

   nextNode = new Node();
   nextNode->data = 66;
```

```
    linkListOne->next = nextNode;
    linkListTwo->next->next->next = nextNode;

    nextNode = new Node();
    nextNode->data = 6;
    linkListOne->next->next = nextNode;

    linkListOne->next->next->next = NULL;

    Node* intersectionCheck = intersectionNode(linkListOne, linkListTwo);

    if (intersectionCheck)
            cout << "Intersection Node: "<< intersectionCheck->data << endl;
    else
           cout << " No Intersection Node"<<endl;

}
```

O/P :



Q 4:

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

int main()
{
        int x;
        cout<<"Enter Array Size :";
        cin>>x;

        int arrx[x-1];

        cout<<"Enter Array Elements :";
        for(int i=0;i<x-1;i++)
        cin>>arrx[i];

        int sum = (x*(x+1))/2;

        for(int i=0;i<x-1;i++)
        sum= sum-arrx[i];

                cout<<"Missing Num:"<<sum;


}
```

O/P :

```
PS F:\DSALAB#> cd "f:\DSALAB#"
PS F:\DSALAB#> cd "f:\DSALAB#\" ; if ($?) { g++ test.cpp -o test } ; if ($?) { .\test }
Enter Array Size :5
Enter Array Elements :1
2
3
5
Missing Num:4
PS F:\DSALAB#>
```

Q5 :

```
#include<iostream>
using namespace std;
class Node{
	public:
	int data;
	Node*next;

	Node(int val){
	data=val;
	next=NULL;
	}
};

void pb(Node* &head,int val){
	Node* n=new Node(val);
	if (head==NULL)
	{
	head = n;
	return;
	}

	Node* temp=head;
	while (temp->next!=NULL)
	{
	temp=temp->next;
	}
	temp->next=n;
}

void sw(Node* head){
	Node* temp=head;
	while (temp!=NULL)
	{
	cout<<temp->data<<"->";
	temp=temp->next;
	}
	cout<<endl;
}

Node* merge(Node* &head1,Node* &head2){

	Node* dummyNode= new Node(-1);
```

```
        Node* pointOne=head1;
        Node* pointTwo=head2;
        Node* pointThree=dummyNode;

        while (pointOne!=NULL && pointTwo!=NULL)
        {
        if (pointOne->data<pointTwo->data)
        {
        pointThree->next=pointOne;
        pointOne=pointOne->next;
        }
        else{
        pointThree->next=pointTwo;
        pointTwo=pointTwo->next;
        }
        pointThree=pointThree->next;
        }
        while (pointOne!=NULL)
        {
        pointThree->next=pointOne;
        pointOne=pointOne->next;
        pointThree=pointThree->next;
        }
        while (pointTwo!=NULL)
        {
        pointThree->next=pointTwo;
        pointTwo=pointTwo->next;
        pointThree=pointThree->next;
        }
 return dummyNode->next;

}
int main(){
        Node* headLL1=NULL;
        Node* headLL2=NULL;

        int x,z,y;
        cout << "Enter LL 1 Size :";
        cin >> x;

        int arr[x];
        for (int i = 0; i < x; i++)
        {
        cin>>z;
```

```
        arr[i] = z;
        }
        for (int i = 0; i < x; i++)
        {
        pb(headLL1,arr[i]);
        }


        cout << "Enter LL 2 Size :";
        cin >> y;
        int arr2[y];

        for (int i = 0; i < y; i++)
        {
        cin>>z;
        arr2[i] = z;
        }
        for (int j = 0; j < y; j++)
        {
        pb(headLL2,arr2[j]);
        }


        cout<<"LL 1:"<<endl;
        sw(headLL1);
        cout<<"LL 2:"<<endl;
        sw(headLL2);
        Node *newhead = merge(headLL1,headLL2);
        cout<<"New LL : "<<endl;
        sw(newhead);
        return 0;
}
```

O/P :

```
PS D:\TLE-Level-1\02-09-2022> cd "d:\TLE-Level-1\02-09-2022"
PS D:\TLE-Level-1\02-09-2022> cd "d:\TLE-Level-1\02-09-2022\" ; if ($?) { g++ sd.cpp -o sd } ; if ($?) { .\sd }
Enter LL 1 Size :4
1
3
5
7
Enter LL 2 Size :4
2
4
6
8
LL 1:
1->3->5->7->
LL 2:
2->4->6->8->
New LL :
1->2->3->4->5->6->7->8->
```