NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

PROGRAM: SOFTWARE ENGINEERING

DATA STRUCTURES LAB

LAB TASK-05

SUBMITTED BY:

Name: Ahmed Ali

Roll No: 22P-9318

INSTRUCTOR NAME: Sir Saood Sarwar

A DEPARTMENT OF COMPUTER SCIENCE

Q1 CODE:

```
#include<iostream>

using namespace std;


class node

{

	public:

		int data;

		node *prev;

		node *next;

};


void remove(node *head, node *del_nod)		//remove node from dll

{

	if(del_nod==nullptr)

	{

		return;

	}


	if(del_nod->prev!=nullptr)

	{

		del_nod->prev->next=del_nod->next;

	}


	if(del_nod->next!=nullptr)

	{

		del_nod->next->prev=del_nod->prev;

	}
```

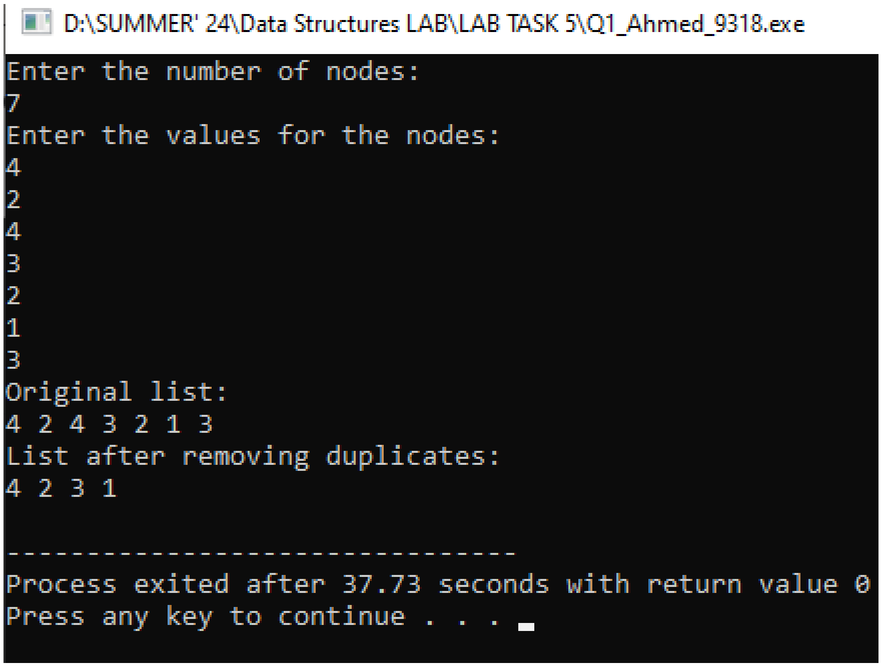
```
    delete del_nod;
}

void remove_duplicates(node *head)     //remove duplicates from dll
{
    if(head==nullptr)
    {
        return;
    }
    node *current=head;
    while(current!=nullptr)
    {
        node *ptr=current->next;

        while(ptr!=nullptr)
        {
            if(ptr->data==current->data)
            {
                node *del_nod=ptr;
                ptr=ptr->next;
                remove(head, del_nod);
            }
            else
            {
                ptr=ptr->next;
            }
        }
```

```
                    current=current->next;
            }
}

void display(node *head)
{
            node *current=head;
            while(current!=nullptr)
            {
                        cout<<current->data;
                        if(current->next!=nullptr)
                        {
          cout<<" ";
                        }
                        current=current->next;
            }
            cout<<endl;
}

// Utility function to insert a new node at the end of the doubly linked list
node *insert_node(node* head, int data)
{
   node *new_node=new node{data};

            if(head==nullptr)
            {
                        head=new_node;
   }
            else
```

```
        {
                node *tail=head;
                while(tail->next!=nullptr)
                {
                        tail=tail->next;
                }
                tail->next=new_node;
                new_node->prev=tail;
        }
        return head;
}

int main()
{
        node *head=nullptr;
        int nodes;
        int value;

        cout<<"Enter the number of nodes: "<<endl;
        cin>>nodes;

        cout<<"Enter the values for the nodes:"<<endl;
  for(int i=0; i<nodes; i++)
        {
    cin>>value;
    head=insert_node(head, value);
  }

        cout<<"Original list:"<<endl;
```

```
	display(head);


	remove_duplicates(head);


	cout<<"List after removing duplicates:"<<endl;
	display(head);


	node *current=head;
	while(current!=nullptr)
	{
		node *temp=current;
		current=current->next;
		delete temp;
	}
	return 0;
}
```

Output-01:



D:\SUMMER' 24\Data Structures LAB\LAB TASK 5\Q1_Ahmed_9318.exe

```
Enter the number of nodes:
7
Enter the values for the nodes:
4
2
4
3
2
1
3
Original list:
4 2 4 3 2 1 3
List after removing duplicates:
4 2 3 1

--------------------------------
Process exited after 37.73 seconds with return value 0
Press any key to continue . . .
```

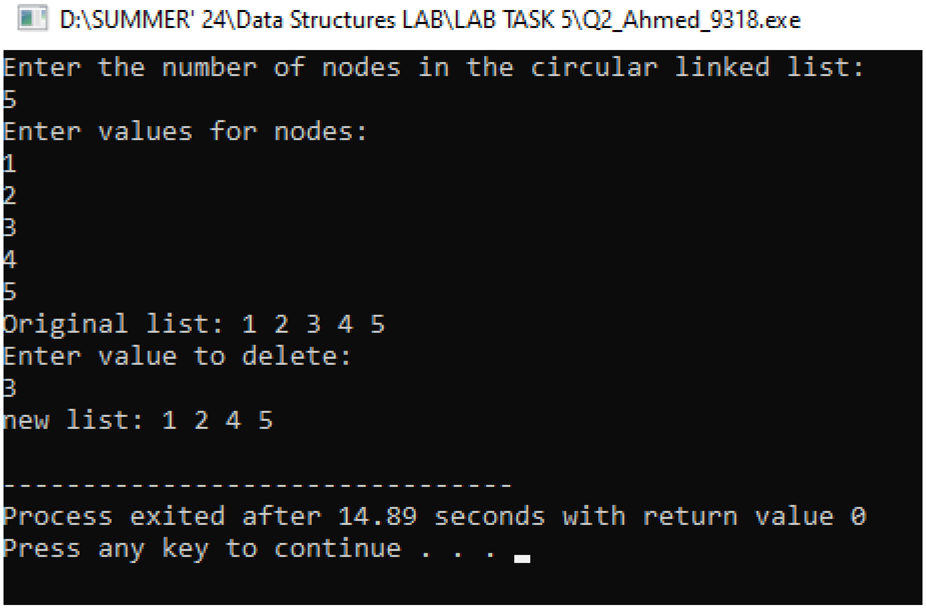
Q2 CODE:

```
#include<iostream>

using namespace std;


class node

{

    public:

        int data;

        node *next;

};


node *remove(node *head, int x)

{

    if(head==nullptr)

    {

        return nullptr;

    }

    node *curr=head;

  node *prev=nullptr;

    do

    {

        if(curr->data==x)

        {

            if(curr==head)
        {
            if(head->next==head)    //deleting head
            {
                delete head;
                return nullptr;
            }
            else
            {
                node *end=head;
                while(end->next!=head)
                {
                    end=end->next;
                }
                head=head->next;
                end->next=head;
                delete curr;
                return head;
            }
        }
        else
        {
            prev->next=curr->next;
            delete curr; //removing nod that isnot head
            return head;
        }
    }
    prev=curr;
    curr=curr->next;
}
```

```
        while(curr!=head);

   return head; //means target not found
}

void display(node *head)
{
        if(head==nullptr)
        {
                cout<<"List is empty"<<endl;
                return;
        }
        node *curr=head;
        do
        {
                cout<<curr->data<<" ";
                curr=curr->next;
        }
        while(curr!=head);
        cout<<endl;
}

node *create(int data)
{
        node *newnode=new node();
        newnode->data=data;
        newnode->next=nullptr;
        return newnode;
}
```

```
node *insert(node *head, int value)        //insert in end
{
        node *newnode=create(value);
        if(head==nullptr)
        {
                newnode->next=newnode;
                return newnode;
        }
        node *curr=head;
        while(curr->next!=head)
        {
                curr=curr->next;
        }
        curr->next=newnode;
        newnode->next=head;
        return head;
}

int main()
{
  node *head=nullptr;
        int n, value, target;

        cout<<"Enter the number of nodes in the circular linked list: "<<endl;
        cin>>n;

        cout<<"Enter values for nodes:"<<endl;
        for(int i=0; i<n; i++)
        {
                cin>>value;
                head=insert(head, value);
        }
        cout<<"Original list: ";
        display(head);

        cout<<"Enter value to delete: "<<endl;
        cin>>target;
        head=remove(head, target);
        cout<<"new list: ";
        display(head);
        return 0;
}
```

Output-02:



D:\SUMMER' 24\Data Structures LAB\LAB TASK 5\Q2_Ahmed_9318.exe

```
Enter the number of nodes in the circular linked list:
5
Enter values for nodes:
1
2
3
4
5
Original list: 1 2 3 4 5
Enter value to delete:
3
new list: 1 2 4 5

--------------------------------
Process exited after 14.89 seconds with return value 0
Press any key to continue . . .
```