Data Structure

Assignment – 3

Name – Ananya Agarwal

Roll no – 102083036

Batch – 2CO14

Questions 1:-

```
#include<iostream>
using namespace std;
struct node
{
	int data;
	node *next;
	node *prev;
};
class l_list
{
	node *head;
	public:
		l_list()
		{
			head=NULL;
		}
		void i_beg();
		void i_end();
		void i_after();
		void i_before();
		void d_specific();
		void display();
		void search();
};
void l_list::i_beg()
{
	int a;
	node *temp=new node;
	cout<<"Enter the value to be inserted : ";
	cin>>a;
	temp->data=a;
	temp->next=temp->prev=temp;
```

```
        if(head == NULL)
        head=temp;
        else
        {
                temp->next=head;
                temp->prev=head->prev;
                temp->prev->next=temp;
                head->prev=temp;
                head=temp;
        }
}
void l_list::i_end()
{
        int a;
        node *temp=new node;
        cout<<"Enter the value to be inserted : ";
        cin>>a;
        temp->data=a;
        temp->next=temp->prev=temp;
        if(head == NULL)
        head=temp;
        else
        {
                temp->next=head;
                temp->prev=head->prev;
                temp->prev->next=temp;
                head->prev=temp;
        }
}
void l_list::i_after()
{
        int a,x;
        cout<<"Enter the value to be inserted : \n";
```

```
        cin>>a;
        cout<<"Enter the element after which you want to insert the value \n";
        cin>>x;
        if(head!=NULL)
        {
        node *temp=head;
        while(temp->next!=head && temp->data!=x)
        temp=temp->next;
        if(temp->data==x)
          {
                node *temp1=new node;
                temp1->data=a;
                temp1->prev=temp;
                temp1->next=temp->next;
                temp->next=temp1;
                temp1->next->prev=temp1;
          }
   }
}
void l_list::i_before()
{
        int a,x;
        cout<<"Enter the value to be inserted : ";
        cin>>a;
        cout<<"Enter the element before which you want to insert the value : ";
        cin>>x;
        node *p;
        if(head == NULL)
        {
                cout<<"First create the list \n";
                return;
        }
        p=head;
```

```
	do
	{
		if(p->data==x)
		break;
		p=p->next;
	}while(p!=head);
	if(p==head && p->data!=x)
	{
		cout<<"item before which insertion is done is not present in the linked list :
\n";
		return;
	}
	node *temp=new node;
	temp->data=a;
	temp->prev=NULL;
	temp->next=NULL;
	node *q=p->prev;
	q->next=temp;
	temp->prev=q;
	temp->next=p;
	p->prev=temp;
	if(p==head)
	head=temp;
}
void l_list::d_specific()
{
	int x;
	cout<<"Enter the element you wanna delete \n";
	cin>>x;
	if(head!=NULL)
	{
	node *temp=head;
	while(temp->next!=head && temp->data!=x)
	temp=temp->next;
```

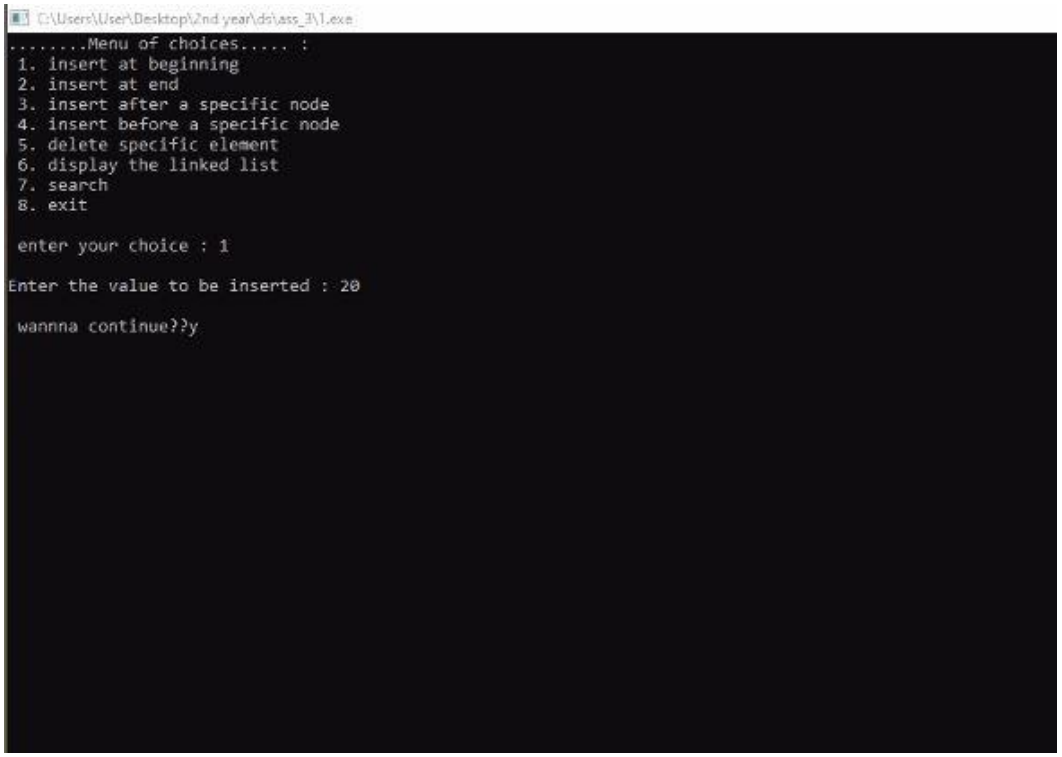
```
        if(temp->data!=x)
        cout<<"not found!! \n";
                else if(temp->next==temp)
                {
                        head=NULL;
                        delete temp;
                        cout<<"......deletion will be successful \n";
                }
                else
                {
                        temp->next->prev=temp->prev;
                        temp->prev->next=temp->next;
                        if(temp==head)
                        head=head->next;
                        delete temp;
                }
   }
}
void l_list::display()
{
    node *ptr;
    ptr=head;
   if(head == NULL)
   {
      cout<<"\n it is an empty list!! \n";
   }
   else
   {
      cout<<"\n the doubly circular linked list elemnts' are : \n";
      while(ptr->next!= head)
      {

         cout<<ptr->data<<' ';
```

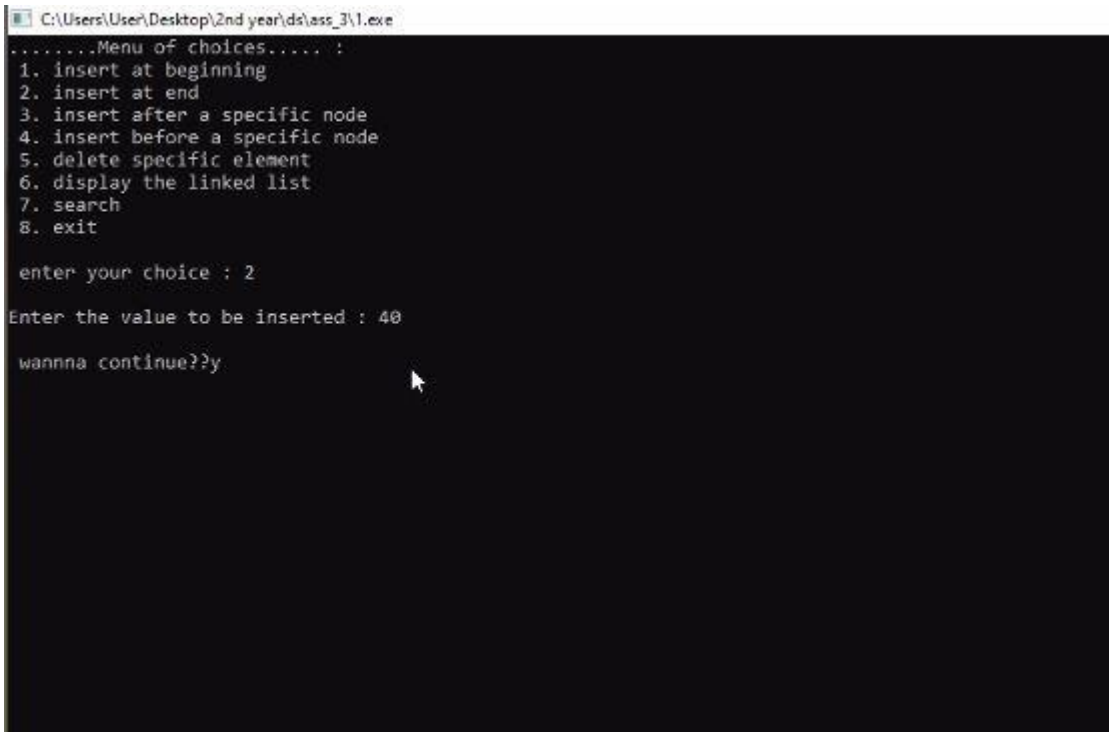
```
            ptr = ptr->next;
        }
        cout<<ptr->data;
    }
}
void l_list::search()
{
        if(head==NULL)
        cout<<"List is empty!!\n";
        else
        {
                int x,c=0;
                node *temp=head;
                cout<<"Enter value of element to be searched : ";
                cin>>x;
                while(temp!=NULL && temp->data!=x)
                {
                        temp=temp->next;
                        c++;
                }
                if(temp==NULL)
                cout<<"Element not found!!!!\n";
                else
                {
                        cout<<temp->data<<" found at position :"<<c+1<<endl;
                }
        }
}
int main()
{
        l_list a;
        char c;
        int ch;
```

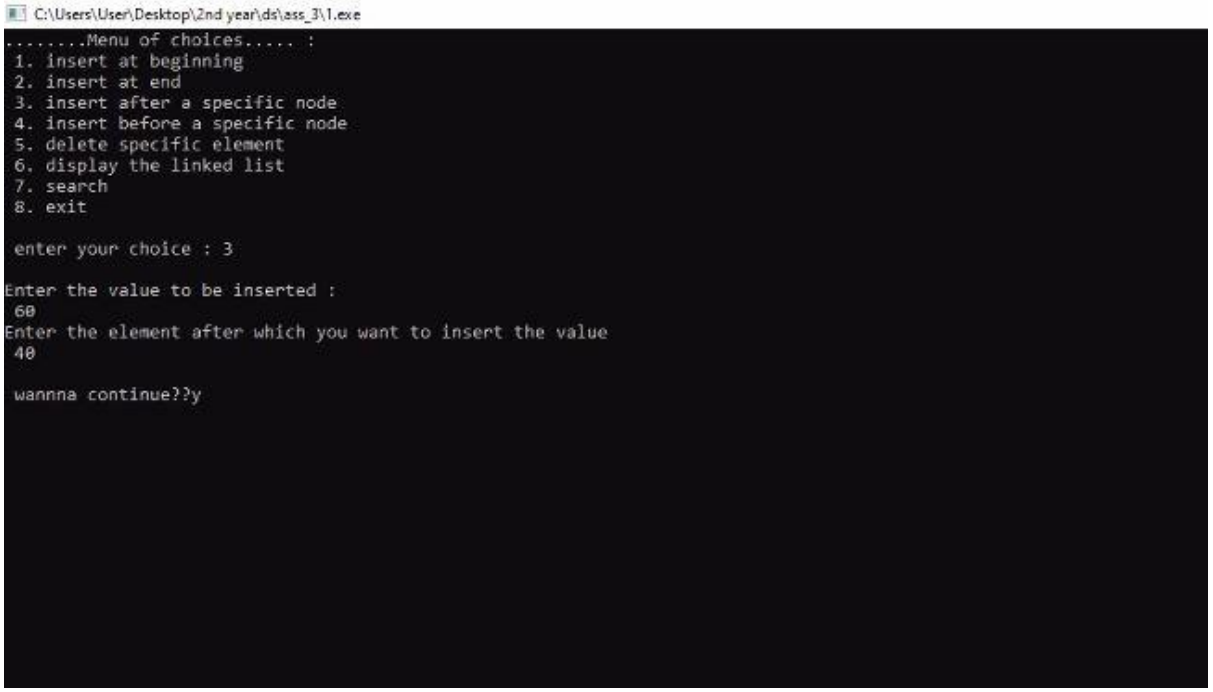
```
do
{
        cout<<"........Menu of choices..... :\n";
        cout<<" 1. insert at beginning \n";
        cout<<" 2. insert at end \n";
        cout<<" 3. insert after a specific node \n";
        cout<<" 4. insert before a specific node \n";
        cout<<" 5. delete specific element \n";
        cout<<" 6. display the linked list \n";
        cout<<" 7. search \n";
        cout<<" 8. exit \n";
        cout<<" \n enter your choice : ";
        cin>>ch;
        cout<<endl;
        switch(ch)
        {
                case 1:
                        a.i_beg();
                        break;
                case 2:
                        a.i_end();
                        break;
                case 3:
                        a.i_after();
                        break;
                case 4:
                        a.i_before();
                        break;
                case 5:
                        a.d_specific();
                        break;
                case 6:
                        a.display();
```

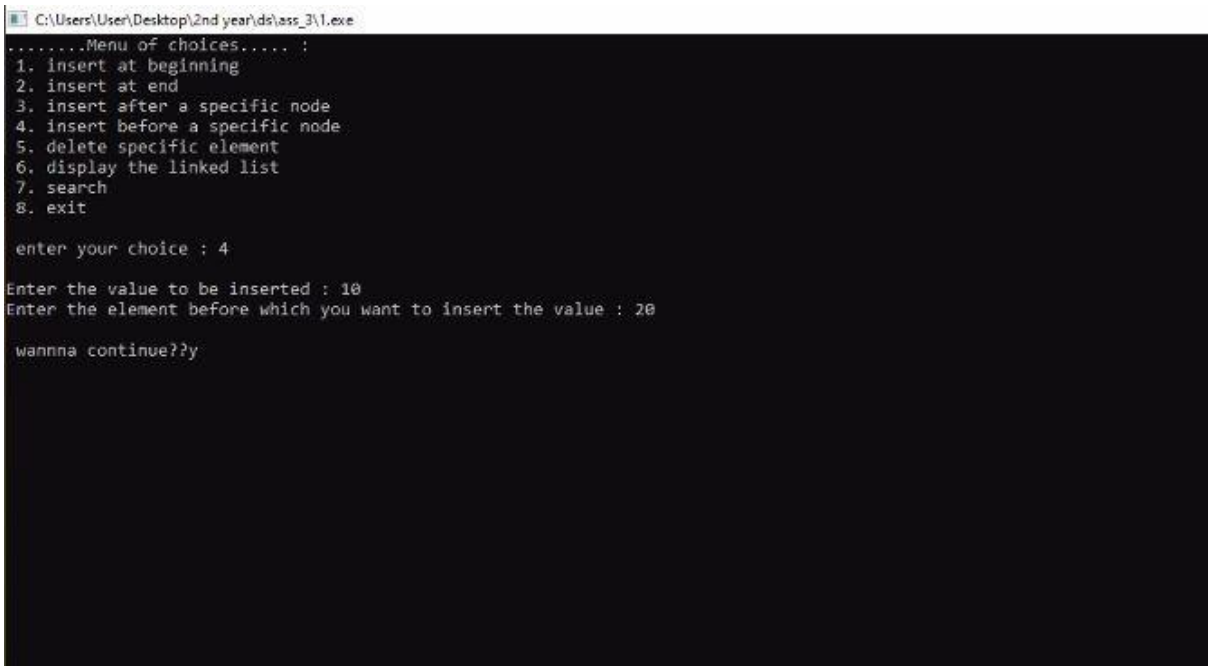
```
                    break;
                case 7:
                    a.search();
                    break;
                case 8:
                    exit(0);
                default:
                cout<<"Invalid choice!!\n";
            }
            cout<<"\n wannna continue??";
            cin.ignore();
            cin>>c;
            system("cls");
        }
        while(c=='y'|| c=='Y');
        }
```

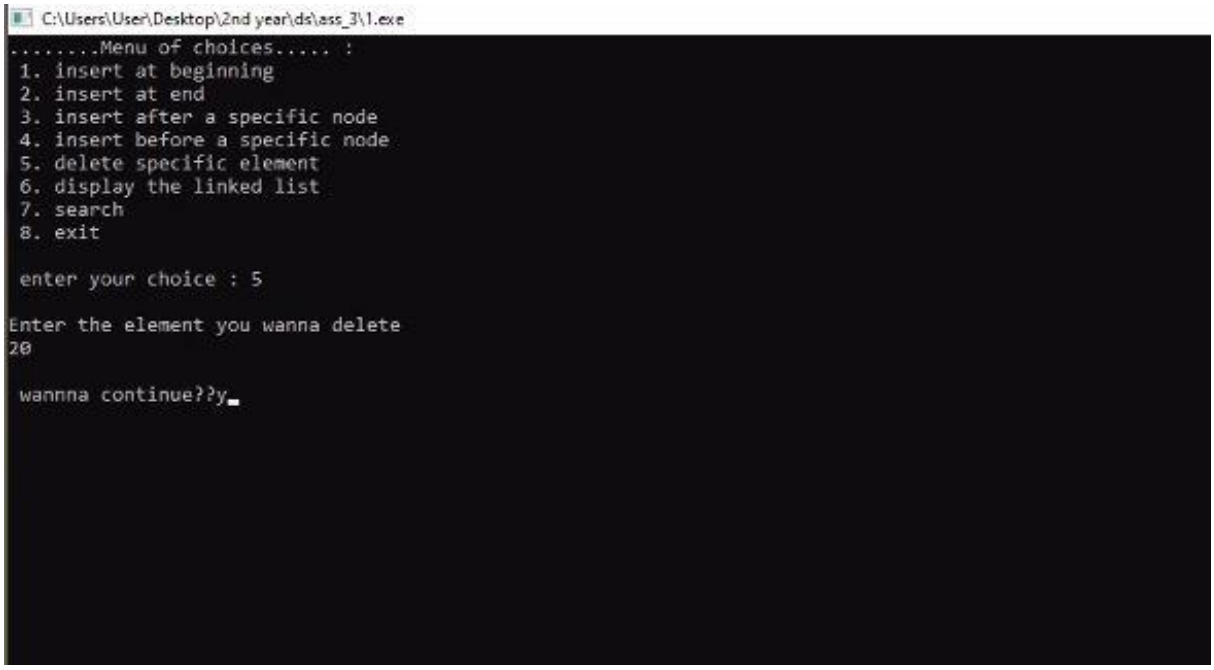
Output:-

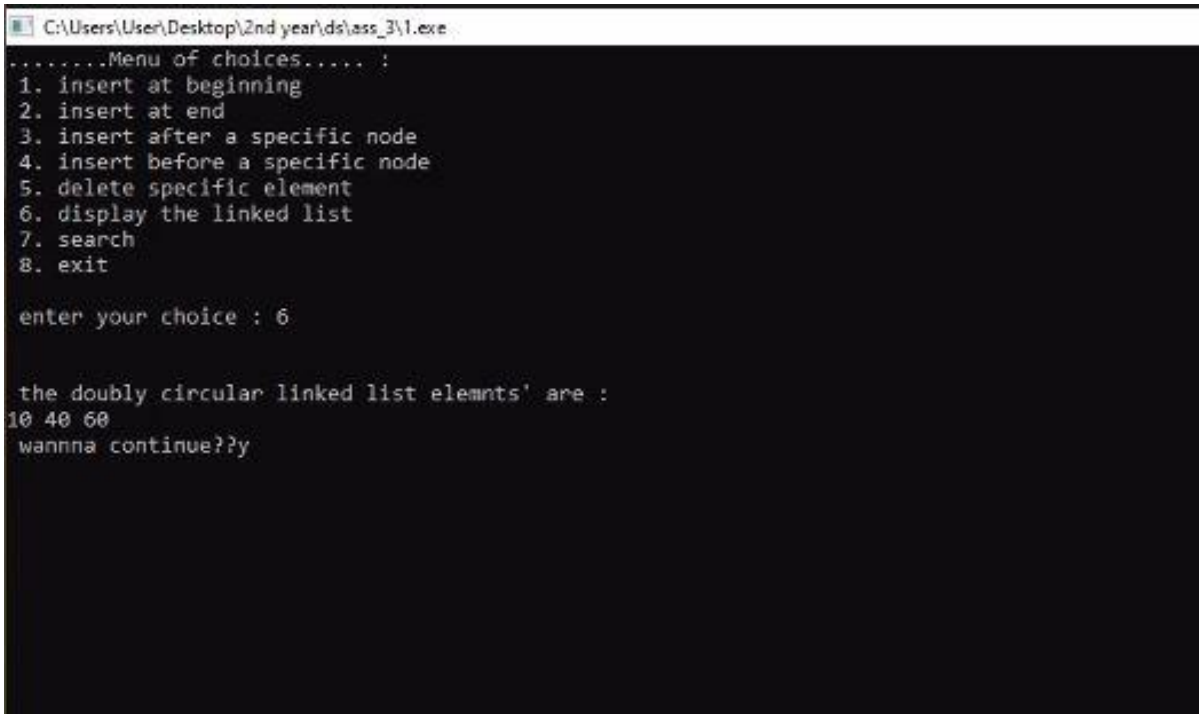


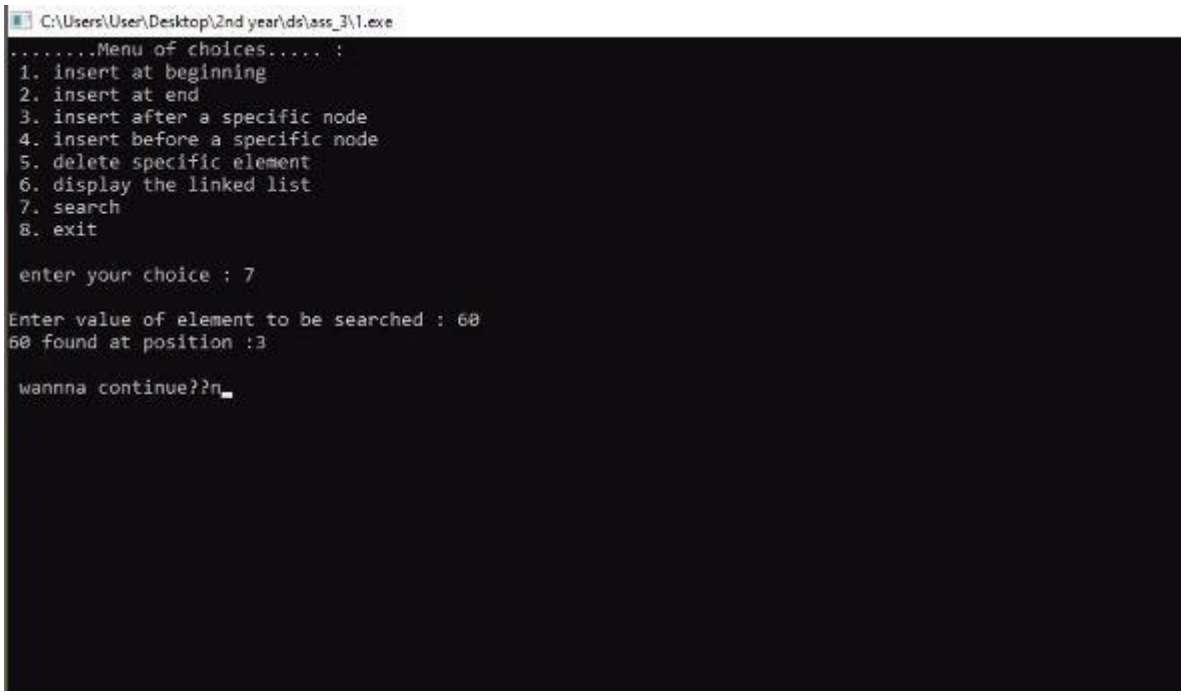

C:\Users\User\Desktop\2nd year\ds\ass_3\1.exe
.....Menu of choices..... :
1. insert at beginning
2. insert at end
3. insert after a specific node
4. insert before a specific node
5. delete specific element
6. display the linked list
7. search
8. exit
enter your choice : 2
Enter the value to be inserted : 40
wannna continue??y


C:\Users\User\Desktop\2nd year\ds\ass_3\1.exe
.....Menu of choices..... :
1. insert at beginning
2. insert at end
3. insert after a specific node
4. insert before a specific node
5. delete specific element
6. display the linked list
7. search
8. exit
enter your choice : 3
Enter the value to be inserted :
60
Enter the element after which you want to insert the value
40
wannna continue??y


C:\Users\User\Desktop\2nd year\ds\ass_3\1.exe
.....Menu of choices..... :
1. insert at beginning
2. insert at end
3. insert after a specific node
4. insert before a specific node
5. delete specific element
6. display the linked list
7. search
8. exit
enter your choice : 4
Enter the value to be inserted : 10
Enter the element before which you want to insert the value : 20
wannna continue??y


C:\Users\User\Desktop\2nd year\ds\ass_3\1.exe
.....Menu of choices..... :
1. insert at beginning
2. insert at end
3. insert after a specific node
4. insert before a specific node
5. delete specific element
6. display the linked list
7. search
8. exit
enter your choice : 5
Enter the element you wanna delete
20
wannna continue??y


C:\Users\User\Desktop\2nd year\ds\ass_3\1.exe
.....Menu of choices..... :
1. insert at beginning
2. insert at end
3. insert after a specific node
4. insert before a specific node
5. delete specific element
6. display the linked list
7. search
8. exit
enter your choice : 6
the doubly circular linked list elemnts' are :
10 40 60
wannna continue??y



Questions 2 :-

```
#include<iostream>
using namespace std;
struct node
{
	int data;
	node *next;
};
class l_list
{
	node *head;
	public:
		l_list()
		{
			head=NULL;
		}
		void i_beg();
		void create();
		void display();
};
void l_list::i_beg()
```

```
{
        node *temp=new node;
        cout<<"Enter the value to be inserted : ";
        cin>>temp->data;
        temp->next=temp;
        if(head == NULL)
        head=temp;
        else
        {
                temp->next=head;
                node *temp1=head;
                while(temp1->next!=head)
                temp1=temp1->next;
                temp1->next=temp;
                head=temp;
        }
}
void l_list::create()
{
        int a;
        node *temp=new node;
        cout<<"Enter the value to be inserted : ";
        cin>>a;
        temp->data=a;
        temp->next=temp;
        if(head == NULL)
        head=temp;
        else
        {
                temp->next=head;
                node *temp1=head;
                while(temp1->next!=head)
                temp1=temp1->next;
```

```
                temp1->next=temp;
        }
}
void l_list::display()
{
        if(head==NULL)
        cout<<"List is empty!!\n";
        else
        {
                cout<<"Linked list elements are \n";
                node *temp=head;
                while (temp->next!=head)
                {
                        cout<<temp->data<<' ';
                        temp=temp->next;
                }
                cout<<temp->data;
                temp=temp->next;
                cout<<' ';
                cout<<temp->data;
        }
}
int main()
{
        l_list a;
        char c;
        int ch;
        do
        {
                cout<<"Menu of choices :\n";
                cout<<" 1. insert at beginning \n";
                cout<<" 2. insert other than at the beginning \n";
                cout<<" 3. display \n";
```

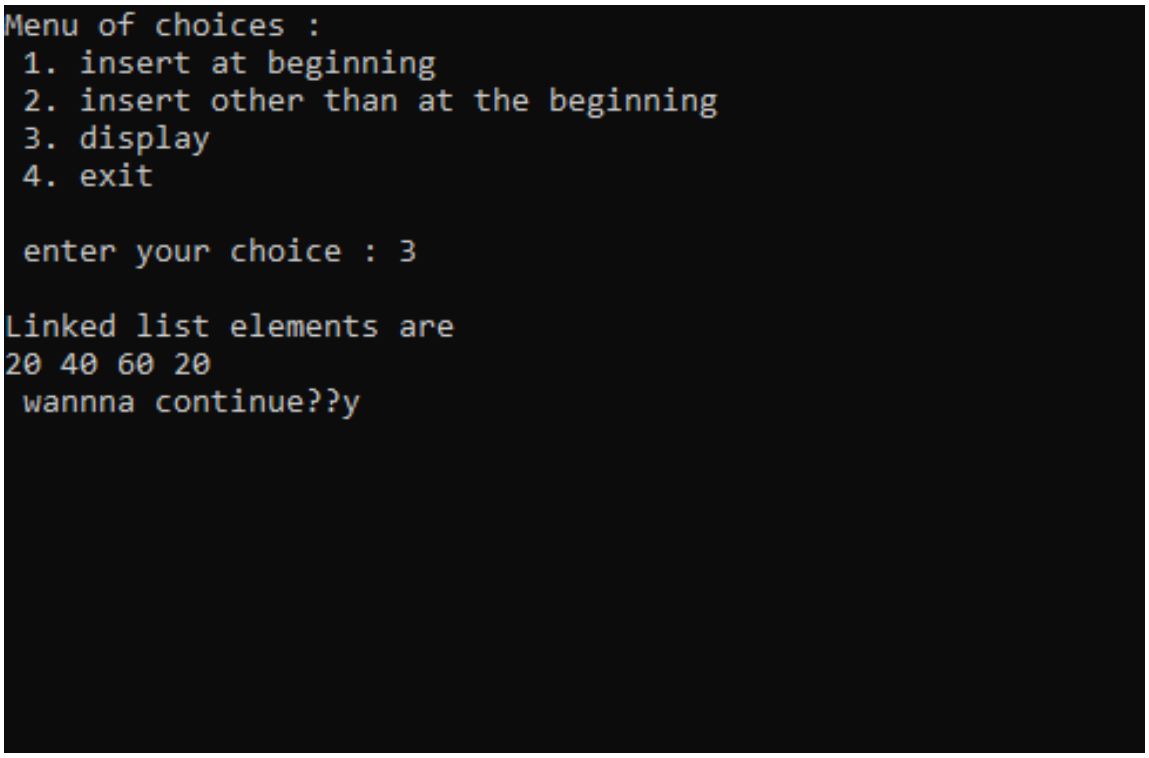
```
        cout<<" 4. exit \n";
        cout<<" \n enter your choice : ";
        cin>>ch;
        cout<<endl;
        switch(ch)
        {
                case 1:
                        a.i_beg();
                        break;
                case 2:
                        a.create();
                        break;
                case 3:
                        a.display();
                        break;
                case 4:
                        exit(0);
                default:
                cout<<"Invalid choice!!\n";
        }
        cout<<"\n wannna continue??";
        cin.ignore();
        cin>>c;
        system("cls");
}
while(c=='y'|| c=='Y');
}
```

Output:-

```
Menu of choices :
 1. insert at beginning
 2. insert other than at the beginning
 3. display
 4. exit

 enter your choice : 1

Enter the value to be inserted : 20

 wannna continue??y
```

```
Menu of choices :
 1. insert at beginning
 2. insert other than at the beginning
 3. display
 4. exit

 enter your choice : 2

Enter the value to be inserted : 40

 wannna continue??y
```

```
Menu of choices :
 1. insert at beginning
 2. insert other than at the beginning
 3. display
 4. exit

 enter your choice : 2

Enter the value to be inserted : 60

 wannna continue??y
```



Questions 3 :-

a)

```
#include<iostream>
using namespace std;
struct node
{
        int data;
        node *next;
        node *prev;
};
class l_list
{
        node *head;
        public:
                l_list()
                {
                        head=NULL;
                }
                void i_beg();
                void create();
                void display();
```

```
                void count();
};
void l_list::i_beg()
{
        node *temp=new node;
        cout<<"Enter the value to be inserted-";
        cin>>temp->data;
        temp->next=temp->prev=NULL;
        if(head == NULL)
        head=temp;
        else
        {
           head->prev=temp;
           temp->next=head;
                head=temp;
        }
}
void l_list::create()
{
                int a;
                node *temp=new node;
                cout<<"Enter the value to be inserted-";
                cin>>a;
                temp->data=a;
                temp->next=temp->prev=NULL;
                if(head == NULL)
                {
                        head=temp;
                }
                else
                {
                node *temp1=head;
                while(temp1->next!=NULL)
                temp1=temp1->next;
                temp1->next=temp;
```

```
                temp->prev=temp1;
        }
}
void l_list::display()
{
        if(head==NULL)
        cout<<"List is empty!!\n";
        else
        {
                cout<<"Linked list elements : \n";
                node *temp=head;
                while(temp!=NULL)
                {
                        cout<<temp->data<<' ';
                        temp=temp->next;
                }
                cout<<endl;
        }
}
void l_list::count()
{
        node *temp=head;
        int c=0;
        while(temp!=NULL)
        {
                c++;
                temp=temp->next;
        }
        cout<<c;
}
int main()
{
        l_list a;
        char c;
        int ch;
```

```
	do
	{
		cout<<"Menu of choices :\n";
		cout<<" 1. insert at beginning \n";
		cout<<" 2. insert at other than at the beginning \n";
		cout<<" 3. display \n";
		cout<<" 4. to get the count of the total number of nodes present int the doubly
linked list \n";
		cout<<" 5. exit \n";
		cout<<" \n enter your choice : ";
		cin>>ch;
		cout<<endl;
		switch(ch)
		{
			case 1:
				a.i_beg();
				break;
			case 2:
				a.create();
				break;
			case 3:
				a.display();
				break;
			case 4:
				a.count();
			case 5:
				exit(0);
			default:
			cout<<"Invalid choice!!\n";
		}
		cout<<"\n wannna continue??";
		cin.ignore();
		cin>>c;
		system("cls");
	}
	while(c=='y'|| c=='Y');   }
```

Output:-

```
Menu of choices :
 1. insert at beginning
 2. insert at other than at the beginning
 3. display
 4. to get the count of the total number of nodes present int the doubly linked list
 5. exit

 enter your choice : 4

2
--------------------------------
Process exited after 69.8 seconds with return value 0
Press any key to continue . . .
```

```
Menu of choices :
 1. insert at beginning
 2. insert at other than at the beginning
 3. display
 4. to get the count of the total number of nodes present int the doubly linked list
 5. exit

 enter your choice : 3

Linked list elements :
20 40

 wannna continue??y
```

```
Menu of choices :
 1. insert at beginning
 2. insert at other than at the beginning
 3. display
 4. to get the count of the total number of nodes present int the doubly linked list
 5. exit

 enter your choice : 2

Enter the value to be inserted-40

 wannna continue??y
```

```
Menu of choices :
 1. insert at beginning
 2. insert at other than at the beginning
 3. display
 4. to get the count of the total number of nodes present int the doubly linked list
 5. exit

 enter your choice : 1

Enter the value to be inserted-20

 wannna continue??y
```

b)

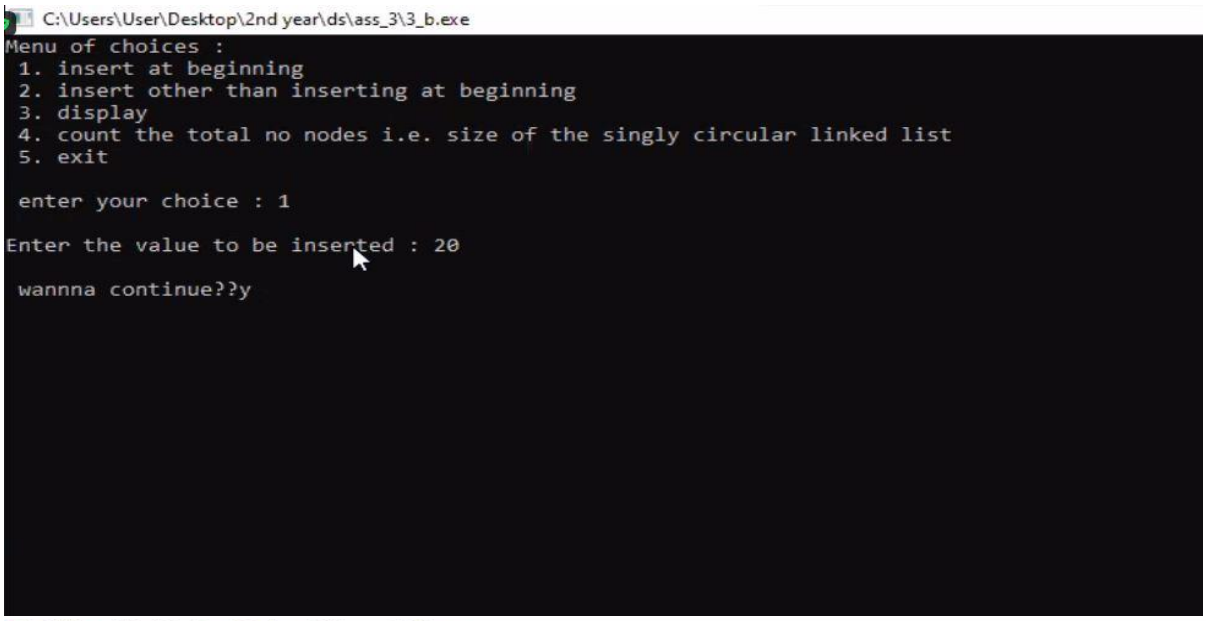
```
#include<iostream>
using namespace std;
struct node
{
	int data;
	node *next;
};
class l_list
{
	node *head;
	public:
		l_list()
		{
			head=NULL;
		}
		void i_beg();
		void create();
		void display();
		void count();
};
void l_list::i_beg()
{
	node *temp=new node;
	cout<<"Enter the value to be inserted : ";
	cin>>temp->data;
	temp->next=temp;
	if(head == NULL)
	head=temp;
	else
	{
		temp->next=head;
		node *temp1=head;
		while(temp1->next!=head)
		temp1=temp1->next;
```

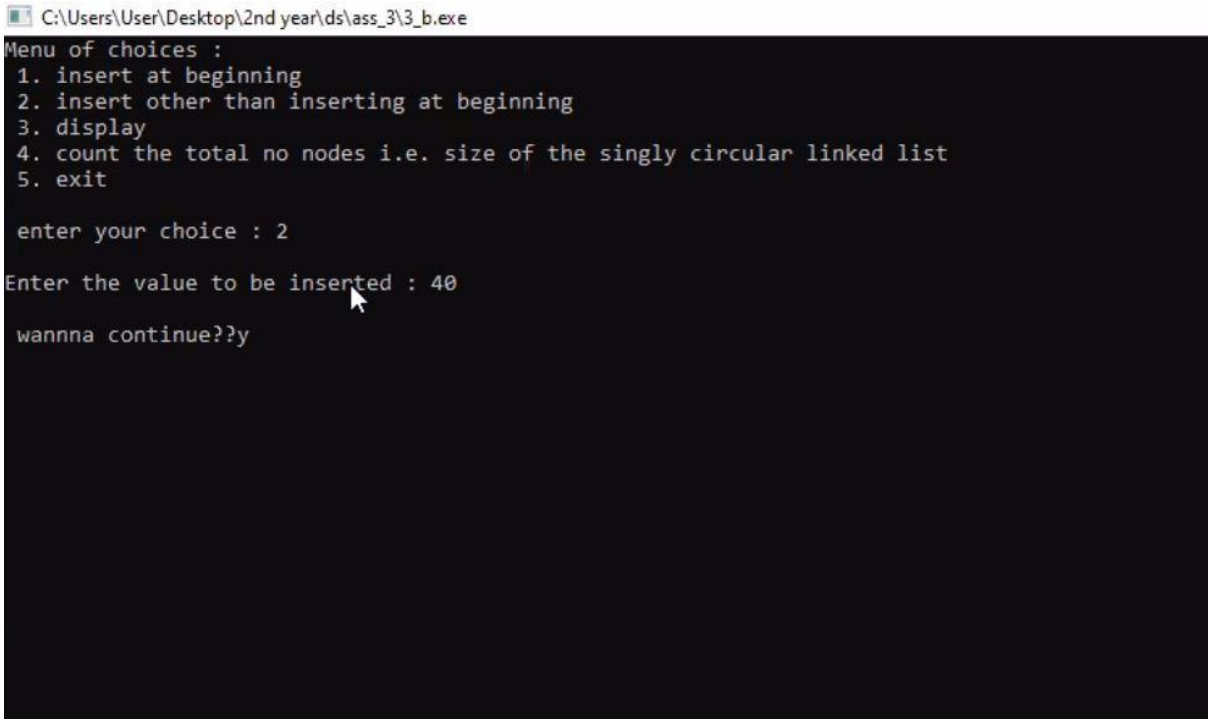
```
                temp1->next=temp;
                head=temp;
        }
}
void l_list::create()
{
        int a;
        node *temp=new node;
        cout<<"Enter the value to be inserted : ";
        cin>>a;
        temp->data=a;
        temp->next=temp;
        if(head == NULL)
        head=temp;
        else
        {
                temp->next=head;
                node *temp1=head;
                while(temp1->next!=head)
                temp1=temp1->next;
                temp1->next=temp;
        }
}
void l_list::display()
{
        if(head==NULL)
        cout<<"List is empty!!\n";
        else
        {
                cout<<"Linked list elements are \n";
                node *temp=head;
                while (temp->next!=head)
                {
                        cout<<temp->data<<' ';
                        temp=temp->next;
```

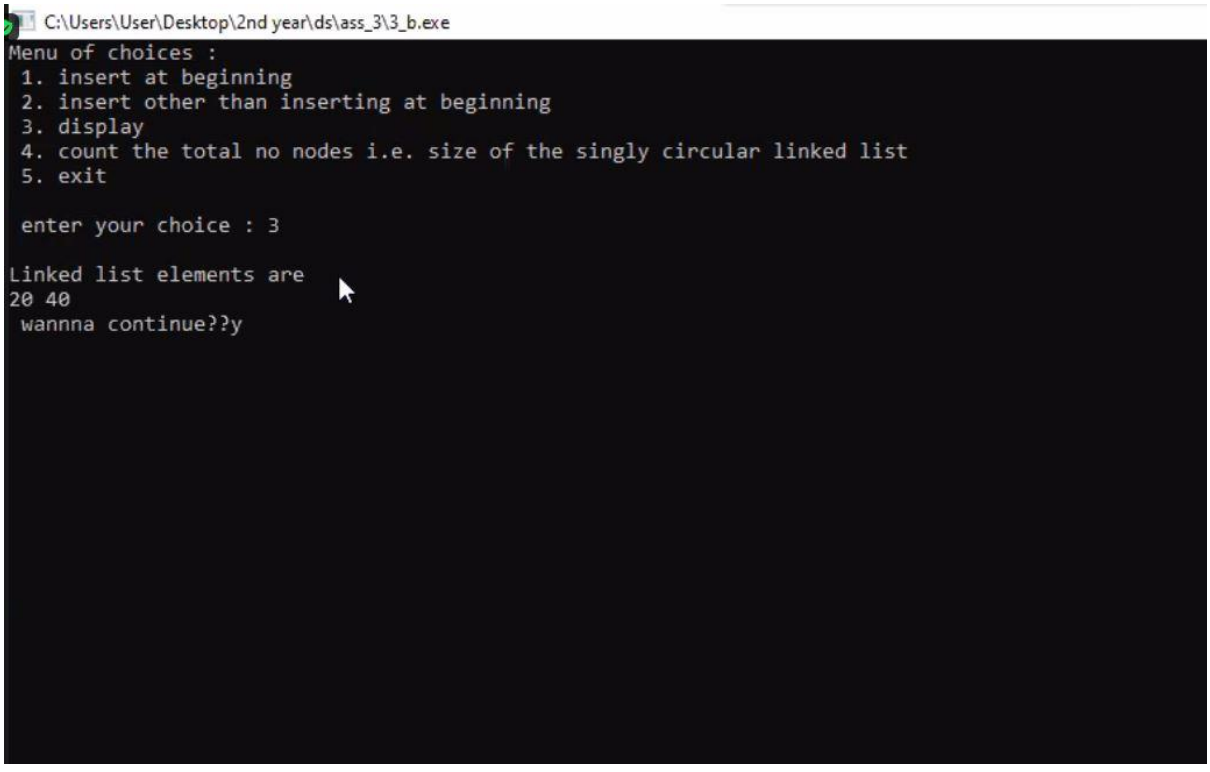
```
                }
                cout<<temp->data;
        }
}
void l_list::count()
{
        node *temp=head;
        int c=0;
        while (temp->next!=head)
        {
                c++;
                temp=temp->next;
        }
        c++;
        cout<<"count="<<c;
}
int main()
{
        l_list a;
        char c;
        int ch;
        do
        {
                cout<<"Menu of choices :\n";
                cout<<" 1. insert at beginning \n";
                cout<<" 2. insert other than inserting at beginning \n";
                cout<<" 3. display \n";
                cout<<" 4. count the total no nodes i.e. size of the singly circular linked list \n";
                cout<<" 5. exit \n";
                cout<<" \n enter your choice : ";
                cin>>ch;
                cout<<endl;
                switch(ch)
                {
                        case 1:
```

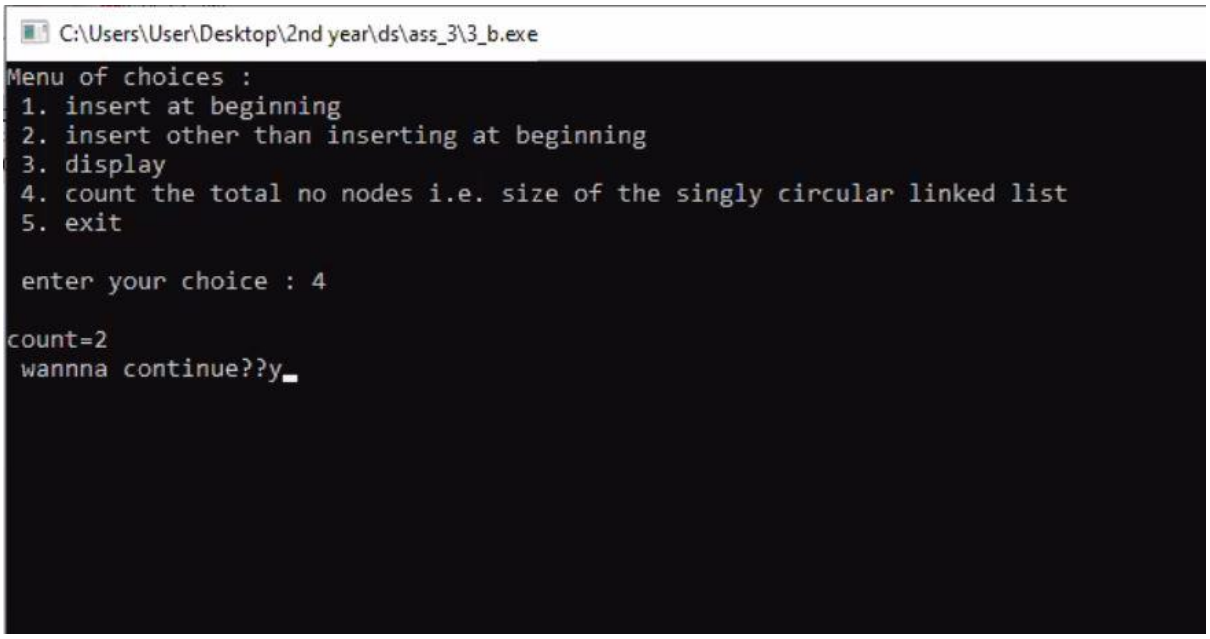
```
                a.i_beg();
                break;
            case 2:
                a.create();
                break;
            case 3:
                a.display();
                break;
            case 4:
                a.count();
                break;
            case 5:
                exit(0);
            default:
            cout<<"Invalid choice!!\n";
        }
        cout<<"\n wannna continue??";
        cin.ignore();
        cin>>c;
        system("cls");
}
while(c=='y'|| c=='Y');
}
```

Output:-


C:\Users\User\Desktop\2nd year\ds\ass_3\3_b.exe
Menu of choices :
1. insert at beginning
2. insert other than inserting at beginning
3. display
4. count the total no nodes i.e. size of the singly circular linked list
5. exit
enter your choice : 1
Enter the value to be inserted : 20
wannna continue??y


C:\Users\User\Desktop\2nd year\ds\ass_3\3_b.exe
Menu of choices :
1. insert at beginning
2. insert other than inserting at beginning
3. display
4. count the total no nodes i.e. size of the singly circular linked list
5. exit
enter your choice : 2
Enter the value to be inserted : 40
wannna continue??y


C:\Users\User\Desktop\2nd year\ds\ass_3\3_b.exe
Menu of choices :
1. insert at beginning
2. insert other than inserting at beginning
3. display
4. count the total no nodes i.e. size of the singly circular linked list
5. exit
enter your choice : 3
Linked list elements are
20 40
wannna continue??y



Questions 4:-

```
#include<iostream>
using namespace std;
struct node
{
        int data;
        node *next;
        node *prev;
};
class l_list
{
        node *head;
        public:
                l_list()
                {
                        head=NULL;
                }
                void i_beg();
                void create();
                void display();
                void check();
};
void l_list::i_beg()
```

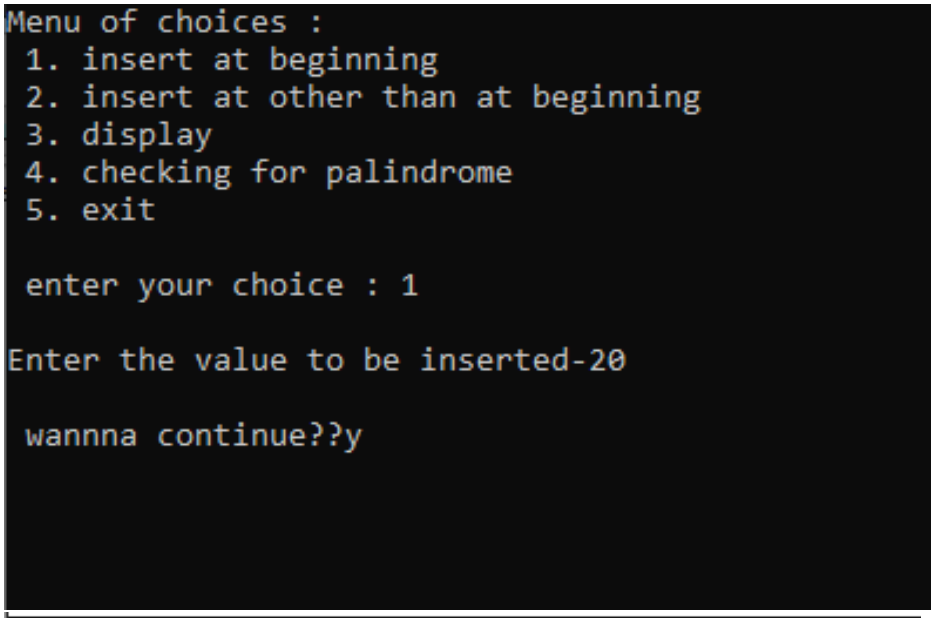
```
{
        node *temp=new node;
        cout<<"Enter the value to be inserted-";
        cin>>temp->data;
        temp->next=temp->prev=NULL;
        if(head == NULL)
        head=temp;
        else
        {
           head->prev=temp;
           temp->next=head;
                head=temp;
        }
}
void l_list::create()
{
                int a;
                node *temp=new node;
                cout<<"Enter the value to be inserted-";
                cin>>a;
                temp->data=a;
                temp->next=temp->prev=NULL;
                if(head == NULL)
                {
                        head=temp;
                }
                else
                {
                node *temp1=head;
                while(temp1->next!=NULL)
                temp1=temp1->next;
                temp1->next=temp;
                temp->prev=temp1;
     }
}
```

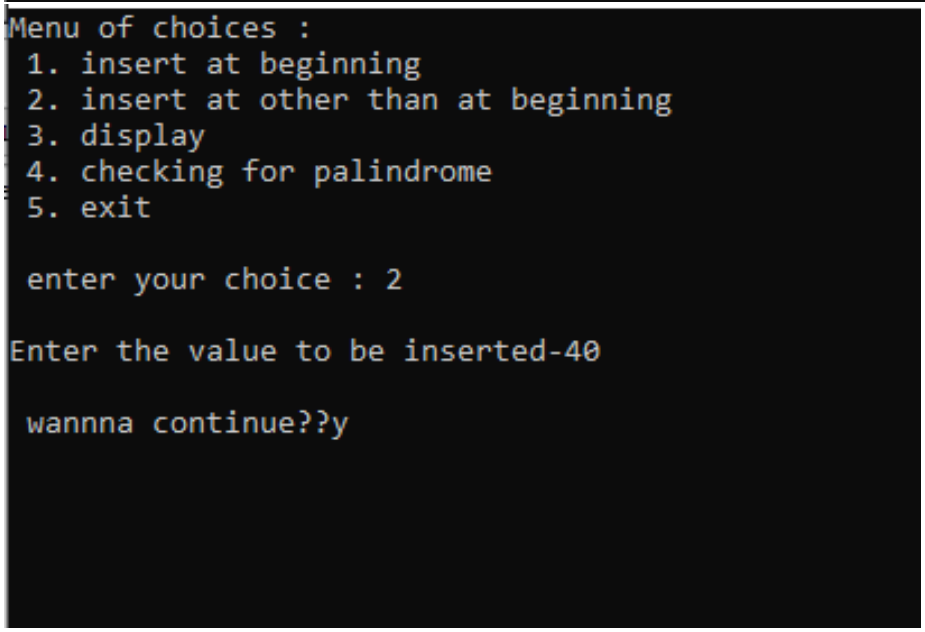
```
void l_list::display()
{
	if(head==NULL)
	cout<<"List is empty!!\n";
	else
	{
		cout<<"Linked list elements : \n";
		node *temp=head;
		while(temp!=NULL)
		{
			cout<<temp->data<<' ';
			temp=temp->next;
		}
		cout<<endl;
	}
}
void l_list::check()
{
	node *temp=head,*tail=head;
	int f=0;
	while(tail->next!=NULL)
	{
		tail=tail->next;
	}
	while (temp!= tail) // now we are at mid since we traversing through temp pointer in forward direction and by pointer tail from backward direction
	{
	if(temp->data!=tail->data)
	 {
		f=1;
		cout<<"\n not palindrome \n";
		break;
	 }
	temp=temp->next; // after we comapre the first and the last , we move 1 towards right through temp and 1 towards left through pointer tail
	tail=tail->prev;
```

```
    }
if (f==0)
cout<<"Yes it is a palindrome!!!!";
}
int main()
{
        l_list a;
        char c;
        int ch;
        do
        {
                cout<<"Menu of choices :\n";
                cout<<" 1. insert at beginning \n";
                cout<<" 2. insert at other than at beginning \n";
                cout<<" 3. display \n";
                cout<<" 4. checking for palindrome \n";
                cout<<" 5. exit \n";
                cout<<" \n enter your choice : ";
                cin>>ch;
                cout<<endl;
                switch(ch)
                {
                        case 1:
                                a.i_beg();
                                break;
                        case 2:
                                a.create();
                                break;
                        case 3:
                                a.display();
                                break;
                        case 4:
                                a.check();
                                break;
                        case 5:
```

```
                    exit(0);
                default:
                cout<<"Invalid choice!!\n";
            }
            cout<<"\n wannna continue??";
            cin.ignore();
            cin>>c;
            system("cls");
        }
        while(c=='y'|| c=='Y');
        }
```

Output:-





```
Menu of choices :
 1. insert at beginning
 2. insert at other than at beginning
 3. display
 4. checking for palindrome
 5. exit

 enter your choice : 2

Enter the value to be inserted-20

 wannna continue??y
```

```
Menu of choices :
 1. insert at beginning
 2. insert at other than at beginning
 3. display
 4. checking for palindrome
 5. exit

 enter your choice : 3

Linked list elements :
20 40 20

 wannna continue??
```

```
Menu of choices :
 1. insert at beginning
 2. insert at other than at beginning
 3. display
 4. checking for palindrome
 5. exit

 enter your choice : 4

Yes it is a palindrome!!!!
 wannna continue??y
```

Questions 5:-

```
#include<iostream>
using namespace std;
struct node
{
	int data;
	node *next;
};
class l_list
{
	node *head;
	public:
		l_list()
		{
			head=NULL;
		}
		void i_beg();
		void create();
		void display();
		void check();
};
void l_list::i_beg()
{
	node *temp=new node;
	cout<<"Enter the value to be inserted : ";
	cin>>temp->data;
	temp->next=temp;
	if(head == NULL)
	head=temp;
	else
	{
		temp->next=head;
		node *temp1=head;
		while(temp1->next!=head)
		temp1=temp1->next;
```

```
                temp1->next=temp;
                head=temp;
        }
}
void l_list::create()
{
        int a;
        node *temp=new node;
        cout<<"Enter the value to be inserted : ";
        cin>>a;
        temp->data=a;
        temp->next=temp;
        if(head == NULL)
        head=temp;
        else
        {
                temp->next=head;
                node *temp1=head;
                while(temp1->next!=head)
                temp1=temp1->next;
                temp1->next=temp;
        }
}
void l_list::display()
{
        if(head==NULL)
        cout<<"List is empty!!\n";
        else
        {
                cout<<"Linked list elements are \n";
                node *temp=head;
                while (temp->next!=head)
                {
                        cout<<temp->data<<' ';
                        temp=temp->next;
```

```
                }
                cout<<temp->data;
        }
}
void l_list::check()
{
        node *temp=head;
        while (temp->next!=NULL && temp->next!=head)

        /* have checked for both the conditions since let if the linked list was circular
         then the frst condition gets satisfied and an infinite loop will execute and thus the second condition is needed
         the second condition is needed because let say if right now temp->next=null then after temp=temp->next.....now in null->next == head here, there is no significance
         of accessing null->next thus system is crashed thus along with second condition , first condition of getting next != null is also must*/

        {
                temp=temp->next;
                if(temp->next == head)
                cout<<"yes it is a singly circular linked list \n";
                else
                cout<<"not a singly circular linked list \n";
        }
}
int main()
{
        l_list a;
        char c;
        int ch;
        do
        {
                cout<<"Menu of choices :\n";
                cout<<" 1. insert at beginning \n";
                cout<<" 2. insert other than at the beginning \n";
                cout<<" 3. display \n";
                cout<<" 4. check for the existence of the singly circular linkd list \n";
```

```
        cout<<" 5. exit \n";
        cout<<" \n enter your choice : ";
        cin>>ch;
        cout<<endl;
        switch(ch)
        {
                case 1:
                        a.i_beg();
                        break;
                case 2:
                        a.create();
                        break;
                case 3:
                        a.display();
                        break;
                case 4:
                        a.check();
                        break;
                case 5:
                        exit(0);
                default:
                cout<<"Invalid choice!!\n";
        }
        cout<<"\n wannna continue??";
        cin.ignore();
        cin>>c;
        system("cls");
}
while(c=='y'|| c=='Y');
}
```

Output:-

```
Menu of choices :
 1. insert at beginning
 2. insert other than at the beginning
 3. display
 4. check for the existence of the singly circular linkd list
 5. exit

 enter your choice : 1

Enter the value to be inserted : 20

 wannna continue??y
```

```
Menu of choices :
 1. insert at beginning
 2. insert other than at the beginning
 3. display
 4. check for the existence of the singly circular linkd list
 5. exit

 enter your choice : 2

Enter the value to be inserted : 40

 wannna continue??y
```

```
Menu of choices :
 1. insert at beginning
 2. insert other than at the beginning
 3. display
 4. check for the existence of the singly circular linkd list
 5. exit

 enter your choice : 3

Linked list elements are
20 40
 wannna continue??y
```

```
Menu of choices :
 1. insert at beginning
 2. insert other than at the beginning
 3. display
 4. check for the existence of the singly circular linkd list
 5. exit

 enter your choice : 4

yes it is a singly circular linked list

 wannna continue??y
```

Questions 5 (New method as in geeks-for-geeks):-

```
#include<iostream>

using namespace std;

struct Node

{

   int data;

   Node* next;

};

bool isCircular(struct Node *head)

{

   if (head == NULL)

   return true;

   struct Node *node = head->next;

   while (node != NULL && node != head)

    node = node->next;
```

```
    return (node == head);
}
Node *newNode(int data)
{
    struct Node *temp = new Node;
    temp->data = data;
    temp->next = NULL;
    return temp;
}
int main()
{
    struct Node* head = newNode(1);
    head->next = newNode(2);
    head->next->next = newNode(3);
    head->next->next->next = newNode(4);

    isCircular(head)? cout << "Yes...\n" :
    cout << "No.....not circular \n" ;
    head->next->next->next->next = head;
    isCircular(head)? cout << "Yes....circular\n" :
    cout << "No....\n" ;
    return 0;
}
```

Output:-