/\*

***Name* :-** Abrar Mahedavi

***Roll No.* :-** 21035

***Title* :-** Write a program using object oriented programming features to implement doubly circular link list with different manipulating facility in c++.

/\*

***Program :***

#include<iostream>

using namespace std;

class node //Declaration of Class

{

public:

int data;

node \*prev;

node \*next;

node() //Default Constructor

{

prev=next=NULL;

}

};

class dcll //Declaration of Class

{

public:

node \*head;

void create(); //Prototype for create function

void display(); //Prototype for display function

void insert\_beg(); //Prototype for insert\_beg function

void insert\_end(); //Prototype for insert\_end function

void insert\_between(); //Prototype for insert\_between function

void del\_beg(); //Prototype for del\_beg function

void del\_end(); //Prototype for del\_end function

void del\_between(); //Prototype for del\_between function

void search(); //Prototype for search function

dcll() //Constructor Defination

{

head=NULL;

}

};

void dcll::create() //Creating the node

{

int flag=1;

char ans;

node \*p,\*temp;

head=new node;

do

{

p=new node;

p->next=NULL;

p->prev=NULL;

cout<<"\nEnter the data:";

cin>>p->data;

if(flag==1) //Flag for head node

{

head=p;

p->prev=head;

p->next=head;

flag=0; //Resting flag for other nodes

}

else

{

temp=head;

while(temp->next!=head) //Finding last node

temp=temp->next;

p->next=head;

p->prev=temp;

temp->next=p;

head->prev=p;

}

cout<<"\nYou want to enter more nodes?:";

cin>>ans;

}

while(ans=='Y'||ans=='y');

}

void dcll::display() //Function for Displaying node

{

node \*temp;

temp=head;

if(temp==NULL)

cout<<"\n LIST IS EMPTY"; //Check the list is EMPTY or NOT

else

do

{

cout<<temp->data<<"==> ";

temp=temp->next;

}while(temp!=head);

}

void dcll::insert\_beg() //Function for Insert at Start

{

node \*p,\*temp;

p=new node;

p->next=NULL;

p->prev=NULL;

cout<<"\nEnter data:";

cin>>p->data;

if(head==NULL) //If list is EMPTY,take new node as head

{

head=p;

p->prev=head;

p->next=head;

}

else

{

temp=head;

while(temp->next!=head)

temp=temp->next;

p->next=head;

p->prev=temp;

head->prev=p;

temp->next=p;

head=p;

}

cout<<"\nNode is inserted!";

}

void dcll::insert\_end() //Function for Inserting node at end

{

node \*p,\*temp;

p=new node;

p->prev=NULL;

p->next=NULL;

cout<<"\nEnter the data:";

cin>>p->data;

if(head==NULL) //If list is EMPTY,take new node as HEAD

{

head=p;

p->next=head;

p->prev=head;

}

else

{

temp=head;

while(temp->next!=head)

temp=temp->next;

p->next=head;

p->prev=temp;

head->prev=p;

temp->next=p;

temp->next->prev=temp;

}

cout<<"\nNode is inserted!!";

}

void dcll::insert\_between() //Function for Inserting node in Between

{

int element;

node \*p,\*temp;

p=new node;

p->next=NULL;

p->prev=NULL;

cout<<"\nEnter the data:";

cin>>p->data;

if(head==NULL)

{

head=p;

}

else

{

cout<<"\nEnter the element after which you want to insert node:";

cin>>element;

temp=head;

do

{

if(temp->data==element)

{

p->next=temp->next;

(temp->next)->prev=p;

temp->next=p;

p->prev=temp;

return;

}

else

{

temp=temp->next;

}

}while(temp!=head);

}

cout<<"\nNode is inserted!!";

}

void dcll::del\_beg() //Function for deleting the node at Start

{

node \*temp,\*temp1;

temp=head;

temp1=temp->next;

while(temp->next!=head)

{

temp=temp->next;

}

temp->next=temp1;

temp1->prev=temp;

delete head;

head=temp1;

cout<<"\nDeleted the node at the beginning!!";

}

void dcll::del\_end() //Function for Deleting the node at End

{

node \*temp,\*temp1;

temp=head;

while(temp->next!=head)

{

temp=temp->next;

}

temp1=temp->prev;

temp1->next=head;

head->prev=temp1;

delete temp;

temp=temp1;

cout<<"\n node in the last is deleted";

}

void dcll::del\_between() //Function for Deleting the in Between node

{

node \*temp,\*temp1;

int element;

cout<<"\nEnter the data you want to delete:";

cin>>element;

temp=head;

if(temp->data==element)

{

temp=temp->next;

if(temp==temp1)

{

temp=temp1=NULL;

delete head;

cout<<"\n Node is deleted,linked list is empty";

}

else

{

while(temp->next!=head)

temp=temp->next;

temp->next=temp1;

temp1->prev=temp;

head=temp1;

cout<<"\nNode is deleted ";

}

}

else

{

while(temp->next!=head)

{

if((temp->next)->data==element)

{

temp1=temp->next;

temp->next=temp1->next;

(temp1->next)->prev=temp;

delete temp1;

cout<<"\nNode is deleted";

}

else

temp=temp->next;

}

}

}

void dcll::search() //Search Function

{

int element;

cout<<"\nEnter the element you want to search:";

cin>>element;

node \*temp;

int flag;

temp=head;

if(temp==NULL) //Checking for EMPTY list condition

{

cout<<"\nThe list is empty";

}

flag=0;

while(temp->next!=head && flag==0)

{

if(temp->data==element)

{

flag=1;

}

else

{

temp=temp->next;

}

}

if(flag==1)

{

cout<<"\nThe node is present";

}

else

{

cout<<"The node is not present";

}

}

int main()

{

node n;

dcll d; //Object is Created

int c;

do

{

cout<<"\n Enter the option you want to perform with nodes: "<<endl;

cout<<"1.CREATE "<<endl;

cout<<"2.DISPLAY "<<endl;

cout<<"3.INSERT AT BEGINNING "<<endl;

cout<<"4.INSERT AT END "<<endl;

cout<<"5.INSERT IN BETWEEN "<<endl;

cout<<"6.DELETE NODE AT BEGINNING "<<endl;

cout<<"7.DELETE THE NODE AT THE END "<<endl;

cout<<"8.DELETE IN BETWEEN "<<endl;

cout<<"9.SEARCH THE NODE "<<endl;

cout<<"10.EXIT "<<endl;

cin>>c; //function calling

switch(c)

{

case 1:d.create();

break;

case 2:d.display();

break;

case 3:d.insert\_beg();

break;

case 4:d.insert\_end();

break;

case 5:d.insert\_between();

break;

case 6:d.del\_beg();

break;

case 7:d.del\_end();

break;

case 8:d.del\_between();

break;

case 9:d.search();

break;

case 10:break;

}

}while(c!=10);

return 0;

}

***Output :***

compeng-sl2-08@compeng-sl2-08:~/Abrar$ g++ DCLL.cpp

compeng-sl2-08@compeng-sl2-08:~/Abrar$ ./a.out

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

1

Enter the data:**10**

You want to enter more nodes?:**y**

Enter the data:**20**

You want to enter more nodes?:**Y**

Enter the data:**30**

You want to enter more nodes?:n

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

2

**10==> 20==> 30==>**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

3

Enter data:**0**

**Node is inserted!**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

2

**0==> 10==> 20==> 30==>**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

4

Enter the data:**40**

**Node is inserted!!**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

2

**0==> 10==> 20==> 30==> 40==>**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

5

Enter the data:**25**

Enter the element after which you want to insert node:**20**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

2

**0==> 10==> 20==> 25==> 30==> 40==>**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

6

**Deleted the node at the beginning!!**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

2

**10==> 20==> 25==> 30==> 40==>**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

7

**node in the last is deleted**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

2

**10==> 20==> 25==> 30==>**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

8

Enter the data you want to delete:**25**

**Node is deleted**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

2

**10==> 20==> 30==>**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

9

Enter the element you want to search:**20**

**The node is present**

Enter the option you want to perform with nodes:

1.CREATE

2.DISPLAY

3.INSERT AT BEGINNING

4.INSERT AT END

5.INSERT IN BETWEEN

6.DELETE NODE AT BEGINNING

7.DELETE THE NODE AT THE END

8.DELETE IN BETWEEN

9.SEARCH THE NODE

10.EXIT

10