**PROGRAM FOR CDLL:**

**#include** <iostream>

**#include**<cstdlib>

**using** **namespace** std;

**class** cdll

{

**private**:

**int** loc;

**struct** node

{

**int** data;

node \*next;

node \*prev;

}\*head,\*head1;

**public**:

**cdll**()

{

head1=NULL;

head=NULL;

loc=0;

}

**void** **create**();

**void** **create1**();

**void** **insert\_start**();

**void** **insert\_end**();

**void** **insert\_mid**();

**void** **delete\_start**();

**void** **delete\_end**();

**void** **delete\_mid**();

**void** **sort**();

**void** **reverse**();

**void** **search**();

**void** **concat**();

**void** **display**();

};

**void** **cdll::create**()

{

node \*temp;

temp=**new** node;

cin>>temp->data;

**if**(head==NULL)

{

temp->next=temp;

temp->prev=temp;

head=temp;

}

**else** **if**(head->next==head)

{

head->next=temp;

temp->prev=head;

temp->next=head;

head->prev=temp;

}

**else**

{

node \*trav;

trav=head;

**while**(trav->next!=head)

{

trav=trav->next;

}

trav->next=temp;

temp->prev=trav;

temp->next=head;

head->prev=temp;

}

}

**void** **cdll::create1**()

{

node \*temp;

temp=**new** node;

cin>>temp->data;

**if**(head1==NULL)

{

temp->next=temp;

temp->prev=temp;

head1=temp;

}

**else** **if**(head1->next==head1)

{

head1->next=temp;

temp->prev=head1;

temp->next=head1;

head1->prev=temp;

}

**else**

{

node \*trav1;

trav1=head1;

**while**(trav1->next!=head1)

{

trav1=trav1->next;

}

trav1->next=temp;

temp->prev=trav1;

temp->next=head1;

head1->prev=temp;

}

}

**void** **cdll::insert\_start**()

{

node \*temp;

temp=**new** node;

cout<<"Enter data to insert in list\n";

cin>>temp->data;

node \*trav;

trav=head;

**while**(trav->next!=head)

{

trav=trav->next;

}

temp->next=head;

head->prev=temp;

trav->next=temp;

temp->prev=trav;

head=temp;

cout<<temp->data<<" inserted at start\n";

}

**void** **cdll::insert\_end**()

{

node \*temp;

temp=**new** node;

cout<<"Enter data to insert in list\n";

cin>>temp->data;

node \*trav;

trav=head;

**while**(trav->next!=head)

{

trav=trav->next;

}

trav->next=temp;

temp->prev=trav;

temp->next=head;

head->prev=temp;

cout<<temp->data<<" inserted at end";

}

**void** **cdll::insert\_mid**()

{

node \*temp;

temp=**new** node;

cout<<"Enter data to insert in list\n";

cin>>temp->data;

cout<<"\nEnter location after which element to be inserted\n";

cin>>loc;

node \*trav;

trav=head;

**for**(**int** i=1;i<loc;i++)

{

trav=trav->next;

}

temp->next=trav->next;

temp->next->prev=temp;

trav->next=temp;

temp->prev=trav;

cout<<temp->data<<" inserted after"<<loc<<"location"<<**endl**;

}

**void** **cdll::delete\_start**()

{

node \*temp;

temp=head;

node \*trav;

**if**(head==NULL)

{

cout<<"List is empty,cannot delete\n";

}

**else** **if**(head->next==head)

{

**delete** temp;

head=NULL;

}

**else**

{

//trav=head;

trav=head->prev;

head=head->next;

head->prev=temp->prev;

trav->next=head;

cout<<temp->data<<" is deleted from start\n";

**delete** temp;

}

}

**void** **cdll::delete\_end**()

{

node \*temp;

temp=head;

node \*trav;

**if**(head==NULL)

{

cout<<"List is empty,cannot delete\n";

}

**else** **if**(head->next==head)

{

**delete** temp;

head=NULL;

}

**else**

{

trav=head;

temp=head->prev;

trav=head->prev;

**if**(temp->next==head)

{

trav=trav->prev;

trav->next=head;

head->prev=trav;

}

cout<<temp->data<<" is deleted from end\n";

**delete** temp;

}

}

**void** **cdll::delete\_mid**()

{

**int** loc;

node \*temp;

temp=head;

node \*temp1;

temp1=head;

node \*trav;

**if**(head==NULL)

{

cout<<"List is empty,cannot delete\n";

}

**else** **if**(head->next==head)

{

**delete** temp;

head=NULL;

}

**else**

{

trav=head;

cout<<"Enter loc at which node is to be deleted\n";

cin>>loc;

**for**(**int** i=0;i<loc;i++)

{

temp=temp->next;

temp1=temp1->next;

trav=trav->next;

}

trav=trav->prev;

temp1=temp1->next;

trav->next=temp1;

temp1->prev=trav;

cout<<temp->data<<" is deleted at location "<<loc<<"\n";

**delete** temp;

}

}

**void** **cdll::sort**()

{

**int** value;

node \*temp;

node \*temp1;

temp=head;

**while**(temp->next!=head)

{

temp1=temp->next;

**while**(temp1!=head)

{

**if**(temp->data>temp1->data)

{

value=temp->data;

temp->data=temp1->data;

temp1->data=value;

}

temp1=temp1->next;

}

temp=temp->next;

}

node \*trav;

trav=head;

cout<<"Sorted list is";

**do**

{

cout<<"\t"<<trav->data;

trav=trav->next;

}**while**(trav!=head);

cout<<"\n";

}

**void** **cdll::reverse**()

{

node \*trav;

trav=head;

trav=head->prev;

cout<<"Reversed list is";

**while**(trav!=head)

{

cout<<"\t"<<trav->data;

trav=trav->prev;

**if**(trav==head)

{

cout<<"\t"<<trav->data;

}

}

cout<<"\n";

}

**void** **cdll::concat**()

{

node \*trav;

trav=head;

trav=head->prev;

node \*trav1;

trav1=head1;

trav1=head1->prev;

trav->next=head1;

head1->prev=trav;

trav1->next=head;

head->prev=trav1;

trav=head;

cout<<"Concatenated list is";

**do**

{

cout<<" "<<trav->data<<" ";

trav=trav->next;

}**while**(trav!=head);

cout<<"\n";

}

**void** **cdll::search**()

{

**int** loc=1,value;

**bool** flag=**false**;

node \*temp;

cout<<"Enter the value to be searched\n";

cin>>value;

temp=head;

**while**(temp->next!=head)

{

**if**(temp->data==value)

{

cout<<"Element found at location "<<loc<<**endl**;

flag=**true**;

}

loc++;

temp=temp->next;

}

**if**(flag!=**true**)

{

cout<<"Element not found\n";

}

}

**void** **cdll::display**()

{

node \*trav;

trav=head;

**if**(head==NULL)

{

cout<<**endl**<<"List is empty,nothing to display\n";

}

cout<<"Linked list is";

**do**

{

cout<<"\t"<<trav->data;

trav=trav->next;

}**while**(trav!=head);

cout<<"\n";

}

**int** **main**()

{

cdll c;

**int** n;

cout<<"Enter no of elements in list\n";

cin>>n;

cout<<"Enter elements in list\n";

**for**(**int** i=0;i<n;i++)

{

c.create();

}

**int** ch,ch1;

**while**(1)

{

cout<<"---------------------------------------"<<**endl**;

cout<<"Menu\n1.Insert\t2.Delete\n3.Display\t4.Search\n5.Sort\t\t6.Reverse\n7.Concat\t8.Create new list\n9.Exit\n";

cout<<"---------------------------------------"<<**endl**;

cout<<"Enter your choice\n";

cin>>ch;

**switch**(ch)

{

**case** 1:

cout<<"\n1.Insert at start\n2.Insert at end\n3.Insert in mid\n";

cout<<"Enter your choice\n";

cin>>ch1;

**switch**(ch1)

{

**case** 1:

c.insert\_start();

**break**;

**case** 2:

c.insert\_end();

**break**;

**case** 3:

c.insert\_mid();

**break**;

}

**break**;

**case** 2:

cout<<"\n1.Delete from start\n2.Delete from end\n3.Delete from mid\n";

cout<<"Enter your choice\n";

cin>>ch1;

**switch**(ch1)

{

**case** 1:

c.delete\_start();

**break**;

**case** 2:

c.delete\_end();

**break**;

**case** 3:

c.delete\_mid();

**break**;

}

**break**;

**case** 3:

c.display();

**break**;

**case** 4:

c.search();

**break**;

**case** 5:

c.sort();

**break**;

**case** 6:

c.reverse();

**break**;

**case** 7:

cout<<"Enter elements of second list\n";

**for**(**int** i=0;i<n;i++)

{

c.create1();

}

c.concat();

**break**;

**case** 8:

**int** n;

cout<<"Enter no. of elements\n";

cin>>n;

cout<<"Enter elements in list\n";

**for**(**int** i=0;i<n;i++)

{

c.create();

}

**break**;

**case** 9:

**exit**(0);

**default**:

cout<<"Invalid choice\n"<<**endl**;

**break**;

}

}

**return** 0;

}

**OUTPUT:**

Enter no of elements in list

5

Enter elements in list

11

66

55

33

44

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

1

1.Insert at start

2.Insert at end

3.Insert in mid

Enter your choice

1

Enter data to insert in list

10

10 inserted at start

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

1

1.Insert at start

2.Insert at end

3.Insert in mid

Enter your choice

2

Enter data to insert in list

20

20 inserted at end---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

1

1.Insert at start

2.Insert at end

3.Insert in mid

Enter your choice

3

Enter data to insert in list

25

Enter location after which element to be inserted

2

25 inserted after2location

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

3

Linked list is 10 11 25 66 55 33 44 20

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

2

1.Delete from start

2.Delete from end

3.Delete from mid

Enter your choice

1

10 is deleted from start

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

2

1.Delete from start

2.Delete from end

3.Delete from mid

Enter your choice

2

20 is deleted from end

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

2

1.Delete from start

2.Delete from end

3.Delete from mid

Enter your choice

3

Enter loc at which node is to be deleted

3

55 is deleted at location 3

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

3

Linked list is 11 25 66 33 44

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

4

Enter the value to be searched

25

Element found at location 2

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

5

Sorted list is 11 25 33 44 66

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

6

Reversed list is 66 44 33 25 11

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

7

Enter elements of second list

99

12

74

65

25

Concatenated list is 11 25 33 44 66 99 12 74 65 25

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

8

Enter no. of elements

4

Enter elements in list

1

2

3

4

---------------------------------------

Menu

1.Insert 2.Delete

3.Display 4.Search

5.Sort 6.Reverse

7.Concat 8.Create new list

9.Exit

---------------------------------------

Enter your choice

9