```
#include <iostream>
using namespace std;
struct node
{
    node *prev;
    int data;
    node *next;
};

void insertatbegin(node *&head, int data)
{
    node *temp = new node;
    temp->data = data;
    temp->next = head;
    temp->prev = NULL;
    if (head != NULL)
    {
        head->prev = temp;
    }
    head = temp;
}
void insertatend(node *&head, int data)
{
    node *temp = new node;
    temp->data = data;
    temp->next = NULL;
    temp->prev = NULL;
    if (head == NULL)
    {
        head = temp;
        return;
    }
    node *temp1 = head;
    while (temp1->next != NULL)
    {
        temp1 = temp1->next;
    }
    temp1->next = temp;
    temp->prev = temp1;
}
void insertatpos(node *&head, int data, int pos)
{
    node *temp = new node;
    temp->data = data;
```

```
    temp->next = NULL;
    temp->prev = NULL;
    if (head == NULL)
    {
        head = temp;
        return;
    }
    node *temp1 = head;
    int i = 1;
    while (i < pos - 1)
    {
        temp1 = temp1->next;
        i++;
    }
    temp->next = temp1->next;
    temp->prev = temp1;
    temp1->next->prev = temp;
    temp1->next = temp;
}
void deleteatbegin(node *&head)
{
    if (head == NULL)
    {
        cout << "List is empty" << endl;
        return;
    }
    node *temp = head;
    head = head->next;
    head->prev = NULL;
    delete temp;
}
void deleteatend(node *&head)
{
    if (head == NULL)
    {
        cout << "List is empty" << endl;
        return;
    }
    node *temp = head;
    while (temp->next != NULL)
    {
        temp = temp->next;
    }
    temp->prev->next = NULL;
    delete temp;
}
```

```
void reversedisplay(node *head)
{
    if (head == NULL)
    {
        cout << "List is empty" << endl;
        return;
    }
    node *temp = head;
    while (temp->next != NULL)
    {
        temp = temp->next;
    }
    cout<<"Null";
    while (temp != NULL)
    {
        cout <<" <---> " << temp->data;
        temp = temp->prev;
    }
    cout <<" <---> Null"<<endl;
}
int search(node *head, int data)
{
    if (head == NULL)
    {
        cout << "List is empty" << endl;
        return -1;
    }
    node *temp = head;
    int i = 1;
    while (temp != NULL)
    {
        if (temp->data == data)
        {
            return i;
        }
        temp = temp->next;
        i++;
    }
    return -1;
}
void display(node *head)
{
    if (head == NULL)
    {
        cout << "List is empty" << endl;
        return;
```

```
    }
    node *temp = head;
    cout<<"Null";
    while (temp != NULL)
    {
        cout <<" <---> " << temp->data;
        temp = temp->next;
    }
    cout <<" <---> Null"<<endl;
}

int main()
{
    node *head = NULL;
    cout << "\n1. Insert at begin\n2. Insert at end\n3. Insert at given
position\n4. Deletion at begin\n5. Deletion at end\n6. Reverse Display\n7.
Search\n8. Display\n9. Exit\n";
    int ch, data, pos;
    while (1)
    {
        cout << "Enter your choice: ";
        cin >> ch;
        switch (ch)
        {
        case 1:
            cout << "Enter the data: ";
            cin >> data;
            insertatbegin(head, data);
            display(head);
            break;
        case 2:
            cout << "Enter the data: ";
            cin >> data;
            insertatend(head, data);
            display(head);
            break;
        case 3:
            cout << "Enter the data: ";
            cin >> data;
            cout << "Enter the position: ";
            cin >> pos;
            insertatpos(head, data, pos);
            display(head);
            break;
        case 4:
            deleteatbegin(head);
```

```
            display(head);
            break;
        case 5:
            deleteatend(head);
            display(head);
            break;
        case 6:
            reversedisplay(head);
            break;
        case 7:
            cout << "Enter the data: ";
            cin >> data;
            pos = search(head, data);
            if (pos == -1)
            {
                cout << "Data not found" << endl;
            }
            else
            {
                cout << "Data found at position: " << pos << endl;
            }
            break;
        case 8:
            display(head);
            break;
        case 9:
            cout<<"Exiting..."<<endl;
            return 0;
        default:
            cout << "Invalid choice" << endl;
        }
    }
    return 0;
}
```

OUTPUT:

```
PS C:\Users\aadil\Desktop\CSE\dsalab> cd "c:\Users\aadil\Desktop\CSE\dsalab\" ; if ($?) { g++ program12.cpp

1. Insert at begin
2. Insert at end
3. Insert at given position
4. Deletion at begin
5. Deletion at end
6. Reverse Display
7. Search
8. Display
9. Exit
Enter your choice: 1
Enter the data: 1
Null <---> 1 <---> Null
Enter your choice: 2
Enter the data: 2
Null <---> 1 <---> 2 <---> Null
Enter your choice: 2
Enter the data: 4
Null <---> 1 <---> 2 <---> 4 <---> Null
Enter your choice: 3
Enter the data: 3
Enter the position: 3
Null <---> 1 <---> 2 <---> 3 <---> 4 <---> Null
Enter your choice: 4
Null <---> 2 <---> 3 <---> 4 <---> Null
Enter your choice: 5
Null <---> 2 <---> 3 <---> Null
Null <---> 1 <---> 2 <---> 3 <---> 4 <---> 5 <---> Null
Enter your choice: 6
Null <---> 5 <---> 4 <---> 3 <---> 2 <---> 1 <---> Null
Enter your choice: 7
Enter the data: 3
Data found at position: 3
Enter your choice: 9
Exiting...
PS C:\Users\aadil\Desktop\CSE\dsalab>
```

THANK YOU