```cpp
#include <iostream>
using namespace std;


// creating node
struct Node
{
    int info;
    Node *Link;
};
Node *start = NULL;


// add Nodes in Linked List
void add(int data)
{
    Node *node = new Node();
    node->info = data;
    node->Link = NULL;
```

```
if (start == NULL)
{
    start = node;
}
else
{
    Node *temp;
    temp = start;

    while (temp->Link != NULL)
    {
        temp = temp->Link;
    }

    temp->Link = node;
}
```

```
    cout << "Data added successfuly" << endl;
    system("pause");
}


// insert element at specific node
void insert(int N, int Item)
{
    if (start == NULL)
    {
        cout << "List is empty ";
    }
    else
    {
        Node *ptr = start;
        int flag = 0;
        while (ptr != NULL)
        {
            if (ptr->info == N)
            {
                Node *new_node = new Node();
// creat new node
                Node *New = new_node;
// creat temporary variable to store add of
new node
                New->info = Item;
                New->Link = ptr->Link;
                ptr->Link = New;
                cout << "Item inserted!" <<
endl;
                system("pause");
                flag = 1;
                break;
            }
            else
            {
                ptr = ptr->Link;
            }
        }
        if (flag == 0)
        {
            cout << "Item Not found!" << endl;
            system("pause");
        }
    }
}
// search elements in Linked list
void search()
{
    if (start == NULL)
    {
        cout << "List is empty! " << endl;
        system("pause");
    }
    else
    {
        int key;
        cout << "Enter value to be find from
the list  ";
        cin >> key;
        Node *ptr = start;
        int flag = 0;
        int count = 0;
        while (ptr != NULL)
        {
            count += 1;
            if (ptr->info == key)
            {
                cout << "Value is Founded at
node " << count << endl;
                flag = 1;
                system("pause");
                break;
            }
            else
            {
                ptr = ptr->Link;
                }
            }
            if (flag == 0)
            {
                cout << "Item is Not Found in
List! " << endl;
                system("pause");
            }
        }
}
// delete first node of list
void delete_first()
{
        if (start == NULL)
        {
            cout << "List is empty! " << endl;
            system("pause");
        }
        else
        {
            start = start->Link;
            cout << "Node is deleted! " << endl;
            system("pause");
        }
    }
    // delete last node of list
    void delete_last()
    {
        if (start == NULL)
        {
            cout << "List is empty! " << endl;
            system("pause");
        }
        else if (start->Link == NULL)
        {
            start = NULL;
            cout << "Node is deleted! " << endl;
            system("pause");
    }
    else
    {
        Node *ptr = start;
        Node *temp = start;


        while (ptr->Link != NULL)
        {
            temp = ptr;
            ptr = ptr->Link;
        }
        temp->Link = NULL;
        cout << "Node is deleted! " << endl;
        system("pause");
    }
}
// delete specific node(acourding it's data)
of list
void delete_specific_node()
```

```
{
    int key;
    cout << "Enter data of node to be delete
";
    cin >> key;


    if (start == NULL)
    {
        cout << "List is empty! " << endl;
        system("pause");
    }
    else if (start->info == key)
    {
        start = start->Link;
        cout << "Node is deleted! " << endl;
        system("pause");
    }
    else
    {
        Node *ptr = start;
        Node *temp = start;
        int flag = 0;
        while (ptr->Link != NULL)
        {
            temp = ptr;
            ptr = ptr->Link;
            if (ptr->info == key)
            {
                temp->Link = ptr->Link;
                flag = 1;
                cout << "Node is deleted! " <<
endl;
                system("pause");
                break;
            }
        }
        if (flag == 0)
        {
            cout << "Node is Not Found in
List! " << endl;
            system("pause");
        }
    }
}
// reverse the List


void reverse()
{
    if (start == NULL)
    {
        cout << "List is empty " << endl;
        system("pause");
    }
    else
    {
        Node *prev = NULL;
        Node *ptr = start;
        Node *rev = NULL;


        while (ptr != NULL)
        {
            rev = prev;
            prev = ptr;
            ptr = ptr->Link;
            prev->Link = rev;
        }
        start = prev;
        cout << "List has been reversed! " <<
endl;
        system("pause");
    }
}


// display Link Nodes
void display()
```

```cpp
{
    Node *current_node = start;
    if (start == NULL)
    {
        cout << "No record found!" << endl;
        system("pause");
    }
    else
    {


        cout << "#################==>Record
is<==#################" << endl;
        while (current_node != NULL)
        {
            cout << current_node->info << " ";
            current_node = current_node->Link;
        }
        cout << endl;
        system("pause");
    }
}
int main()
{
    int data;
    int choice;
    bool flag = true;


    while (flag)
    {
        system("cls");
        cout << endl
             << "1: Data Entry" << endl
             << "2: Show Record" << endl
             << "3: Insert at specific Node"
<< endl
             << "4: Search element from List"
<< endl
```

```
                << "5: Delete first node of List"
<< endl
                << "6: Delete Last node of List"
<< endl
                << "7: Delete specific
node(acourding to it's data)" << endl
                << "8: Reverse the List" << endl
                << "9: Exit" << endl
                << "Please Make Your Choice: ";
        cin >> choice;



        switch (choice)
        {
        case 1:
            system("cls");
            cout << "Enter Data: ";
            cin >> data;
            add(data);
            break;
        case 2:
            system("cls");
            display();
            break;
        case 3:
            system("cls");
            int N, Item;
            cout << "Enter the value after 
which new node is to be inserted  ";
            cin >> N;
            cout << "Enter Value to be 
inserted  ";
            cin >> Item;
            insert(N, Item);
            break;
        case 4:
            system("cls");
            search();
```

```
    break;
case 5:
    system("cls");
    delete_first();
    break;
case 6:
    system("cls");
    delete_last();
    break;
case 7:
    system("cls");
    delete_specific_node();
    break;
case 8:
    system("cls");
    reverse();
    break;
case 9:
    flag = false;
```

```
            break;
        default:
            cout << "Wrong Choice";
            break;
        }
    }
}
```