Name :- Siddhi Vinod Pande **Date:-**
Roll No:- 66 **Class:-** SYBCA **Batch:-**
Practical No 5: Implementation of Linear Link List.

```
#include <iostream.h>
#include <conio.h>
class Node {
public:
   int info;
   Node *next;
};

class linklist {
   Node *start, *p, *n;

public:
   linklist() {
      start = NULL;
   }

   void Insert();
   void Delete();
   void Search();
   void Display();
   int count();
   ~linklist();
};

void linklist::Display() {
   if (start == NULL) {
      cout << "\nList is empty!";
   } else {
      cout << "\nList is: ";
      p = start;
      while (p != NULL) {
         cout << p->info << " ";
         p = p->next;
      }
   }
```

```
}

int linklist::count() {
   int c = 0;
   p = start;
   while (p != NULL) {
      c++;
      p = p->next;
   }
   return c;
}

void linklist::Insert() {
   int pos;
   cout << "\nEnter the position: ";
   cin >> pos;

   if (pos > 0 && pos <= count() + 1) {
      n = new Node();
      cout << "\nEnter the item: ";
      cin >> n->info;
      n->next = NULL;

      if (pos == 1) {
         n->next = start;
         start = n;
      } else {
         p = start;
         for (int i = 1; i < pos - 1; i++)
            p = p->next;
         n->next = p->next;
         p->next = n;
      }
   } else {
      cout << "\nPlease enter a valid position";
   }
}

void linklist::Delete() {
   if (start == NULL) {
```

```
        cout << "\nList is underflow (empty)";
    } else {
        int pos, i;
        cout << "\nEnter the position: ";
        cin >> pos;

        if (pos > 0 && pos <= count()) {
            if (pos == 1) {
                n = start;
                start = start->next;
                cout << "\nDeleted number is: " << n->info;
                delete n;
            } else {
                p = start;
                for (i = 1; i < pos - 1; i++)
                    p = p->next;
                n = p->next;
                p->next = n->next;
                cout << "\nDeleted number is: " << n->info;
                delete n;
            }
        } else {
            cout << "\nPlease enter a valid position";
        }
    }
}

void linklist::Search() {
    if (start == NULL) {
        cout << "\nList is underflow (empty)";
    } else {
        int item, pos = 1;
        cout << "\nEnter the number to be searched: ";
        cin >> item;
        p = start;

        while (p != NULL && p->info != item) {
            p = p->next;
            pos++;
        }
```

```
        if (p != NULL && p->info == item) {
            cout << "\nThe item " << item << " is found at position " << pos;
        } else {
            cout << "\nElement not found in the list";
        }
    }
}

linklist::~linklist() {
    p = start;
    while (p != NULL) {
        n = p;
        p = p->next;
        delete n;
    }
}

int main() {
    linklist obj;
    int ch;
    do {
        cout << "\n**********";
        cout << "\n\t MENU";
        cout << "\n\t 1. Insert";
        cout << "\n\t 2. Delete";
        cout << "\n\t 3. Search";
        cout << "\n\t 4. Display";
        cout << "\n\t 5. Exit";
        cout << "\n\t Enter choice: ";
        cin >> ch;

        switch (ch) {
        case 1: obj.Insert(); break;
        case 2: obj.Delete(); break;
        case 3: obj.Search(); break;
        case 4: obj.Display(); break;
        case 5: break;
        default: cout << "\nSorry.. Invalid choice";
        }
```

```
    } while (ch != 5);

    return 0;
}
```

OUTPUT:

```
MENU
1. Insert
2. Delete
3. Search
4. Display
5. Exit

Enter choice: 1
Enter the position: 1
Enter the item: 77

MENU
1. Insert
2. Delete
3. Search
4. Display
5. Exit

Enter choice: 1
Enter the position: 2
Enter the item: 88

MENU
1. Insert
2. Delete
3. Search
4. Display
5. Exit

Enter choice: 1
Enter the position: 3
Enter the item: 99

MENU
1. Insert
2. Delete
3. Search
4. Display
5. Exit
```

Enter choice: 4
List is: 77 88 99

MENU
1. Insert
2. Delete
3. Search
4. Display
5. Exit

Enter choice: 3
Enter the number to be searched: 88
The item 88 is found at position 2

MENU
1. Insert
2. Delete
3. Search
4. Display
5. Exit
5. Exit

Enter choice: 2
Enter the position: 2
Deleted number is: 88