Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Summer, Year:2022), B.Sc. in CSE (Day)

LAB REPORT NO # 05

Course Title: Data Structure Lab

Course Code: CSE 106 Section: PC-213DA

<u>Student Details</u>

Name	ID
Pankaj Mahanto	213902002

Lab Date	**: 10/08/2022**
Submission Date	**: 16/08/2022**
Course Teacher's Name	**: Farhana Akter Sunny**

Farhana Akter Sunny

Senior Professor

Green University of Bangladesh

[For Teachers use only: Don't Write anything inside the box]

Lab Report Status

Marks: . . . . **Signature:** . . . .

Comments: . . . . **Date:** . . . .

1. TITLE OF THE LAB EXPERIMENT [1]

- Implement a C program that is able to insert element at beginning, last and any specific position using linked list.

  Find the specific node of element that is present or not in the singly linked list.

2. OBJECTION [1]

In this problem I will discuss Link List and how it uses?

3. PROCEDURE /ANALYSIS/DESIGN/PSEUDOCODE [2]

Algorithm to insert element beginning:

```
Step 1: if ptr = NULL write overflow go to Step 7
  [end of if]
Step 2: set new_node = ptr
Step 3: set ptr = ptr → next
Step 4: set new_node → data = val
Step 5: set new_node → next = head
Step 6: set head = new_node
Step 7: exit
```

Algorithm to insert element End:

```
Step 1: if ptr = NULL write overflow  go to Step 1
   [end of if]
Step 2: set new_node = ptr
Step 3: set ptr = ptr - > next
Step 4: set new_node - > data = val
Step 5: set new_node - > next = NULL
Step 6: set ptr = head
Step 7: repeat Step 8 while ptr - > next != NULL
Step 8: set ptr = ptr - > next
[end of loop]
Step 9: set ptr - > next = new_node
Step 10: exit
```

Algorithm to insert at any position:

```
Step 1: if ptr = NULL

write overflow
    goto Step 12
   end of if

Step 2: set new_node = ptr
Step 3: new_node → data = val
```

- **Step 4:** set temp = head
- **Step 5:** set i = 0
- **Step 6:** repeat Step 5 and 6 until i<loc< li=""> </loc<>
- **Step 7:** temp = temp → next
- **Step 8:** if temp = NULL

```
write "desired node not present"
     goto Step 12
    end of if
 end of loop
```

- **Step 9:** ptr → next = temp → next
- **Step 10:** temp → next = ptr
- **Step 11:** set ptr = new_node
- **Step 12:** exit

Algorithm to display of link list:

Step 1: if(head==NULL) write empty go to Step 6 otherwise go Step 2

[end of if structure]

Step 2: set n=head and repeat the Step while(n!=NULL)

write the data of the list

Step 3: set item=item+1

Step 4: set n=n->next

Step 5: write total number of item

Step 6: Exit.

Algorithm to search of link list:

- **Step 1:** set ptr = head
- **Step 2:** set i = 0
- **Step 3:** if ptr = NULL

```
  write "empty list"
  goto Step 8
  end of if
```

- **Step 4:** repeat Step 5 to 7 until ptr != NULL
- **Step 5:** if ptr → data = item

```
  write i+1
 end of if
```

- **Step 6:** i = i + 1
- **Step 7:** ptr = ptr → next

  [end of loop]

- **Step 8:** exit

4. IMPLEMENTATION

```c
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
struct node
{
    int data;

    struct node *next;
};
struct node *head = NULL;
// insert here start!!
//->>insert begenning
void insert_beg()
{
    struct node *new_node;
    new_node = (struct node *)malloc(sizeof(struct node));
    if (new_node == NULL)
    {
        printf("\noverflow list\n");
    }
    else
    {
        int v;
        printf("value add beg:\n");
        scanf("%d", &v);
        new_node->data = v;
        if (head == NULL)
        {
            head = new_node;
            new_node->next = NULL;
        }
        else
        {
            new_node->next = head;
            head = new_node;
        }
    }
}
// insert end of the list?
void insert_end()
{
    struct node *new_node, *temp;
    new_node = (struct node *)malloc(sizeof(struct node));
    if (new_node == NULL)
    {
        printf("\noverflow list\n");
    }
    else
    {
        int v;
        printf("value add End:\n");
        scanf("%d", &v);
        new_node->data = v;
        if (head == NULL)
        {
            head = new_node;
            new_node->next = NULL;
        }
        else
        {
            temp = head;
            while (temp->next != NULL)
            {
                temp = temp->next;
            }
            temp->next = new_node;
            new_node->next = NULL;
        }
    }
}
//->>insert Any position of the node.
void insert_at_any_pos()
{
    struct node *new_node, *ptr;
    new_node = (struct node *)malloc(sizeof(struct node));
    if (new_node == NULL)
    {
        printf("\noverflow list\n");
    }
    else
    {
        int v, i = 1, pos;
        printf("\nwhich pos value add:\n");
        scanf("%d", &pos);
        printf("value add any position:\n");
        scanf("%d", &v);
        new_node->data = v;
        if (head == NULL)
        {
            head = new_node;
            new_node->next = NULL;
        }
        else
        {
            ptr = head;
            while (i < pos)
            {
                ptr = ptr->next;
                i++;
            }
            new_node->next = ptr->next;
            ptr->next = new_node;
        }
    }
}

// insert function.
void insert()
{
    int n;

    printf("\n1.InsertBeg\n2.InsertEnd\n3.InsertAtAPos\n\n");
    printf("enter your choice for insert data in link list:");
    scanf("%d", &n);
    switch (n)
    {
    case 1:
        insert_beg();
        break;
    case 2:
        insert_end();
        break;
    case 3:
        insert_at_any_pos();
        break;

    default:
        printf("Invalid choice");
        break;
    }
```

```c
}

// Display or Show function of the node?
void display()
{
    int item = 0;
    struct node *n;
    n = head;
    if (n == NULL)
    {
        printf("\nEmpty Link list\n");
    }
    else
    {
        while (n != NULL)
        {
            printf("%d  ", n->data);
            item++;
            n = n->next;
        }
    }
    printf("\ntotal node is found=%d\n", item);
}
// Search function of the node
void search()
{
    int i = 0, value, found = 0;
    printf("\nwhich value search:\n");
    scanf("%d", &value);
    if (head == NULL)
    {
        printf("\nEmpty Link list\n");
    }
    else
    {
        struct node *temp;
        temp = head;
        while (temp != NULL)
        {
            if (temp->data == value)
            {
                found = 1;
                break;
            }
            i++;
            temp = temp->next;
        }
    }
    if (!found)
    {
        printf("\n Data not found\n");
    }
    else
    {
        printf("\nvalue position is=%d\n", i + 1);
    }
}
// main function of the node.
int main()
{

    int c;
    while (1)
    {
        printf("\n1.Insert\n3.Display\n4.Search\n\n0.Exit\n");
        printf("enter your choice:");
        scanf("%d", &c);
        switch (c)
        {
        case 1:
            insert();
            break;
        case 3:
            display();
            break;
        case 4:
            search();
            break;
        case 0:
            exit(0);
            break;
        default:
            printf("Invalid choice");
            break;
        }
    }
}
```

5. TEST RESULT

Output:

```
1.Insert
3.Display
4.Search
```

```
0.Exit
enter your choice:1

1.InsertBeg
2.InsertEnd
3.InsertAtAPos

enter your choice for insert data in link list:1
value add beg:
7

1.Insert
3.Display
4.Search
0.Exit
enter your choice:3
7
total node is found=1

1.Insert
3.Display
4.Search
0.Exit
enter your choice:1

1.InsertBeg
2.InsertEnd
3.InsertAtAPos

enter your choice for insert data in link list:2
value add End:
4

1.Insert
3.Display
4.Search
0.Exit
enter your choice:3
7  4
total node is found=2

1.Insert
3.Display
4.Search
0.Exit
enter your choice:1

1.InsertBeg
2.InsertEnd
3.InsertAtAPos

enter your choice for insert data in link list:2
value add End:
10

1.Insert
3.Display
4.Search
0.Exit
enter your choice:1

1.InsertBeg
```

```
2.InsertEnd
3.InsertAtAPos

enter your choice for insert data in link list:1
value add beg:
5

1.Insert
3.Display
4.Search
0.Exit
enter your choice:3
5  7  4  10
total node is found=4

1.Insert
3.Display
4.Search
0.Exit
enter your choice:1

1.InsertBeg
2.InsertEnd
3.InsertAtAPos

enter your choice for insert data in link list:3

which pos value add:
2
value add any position:
25

1.Insert
3.Display
4.Search
0.Exit
enter your choice:3
5  7  25  4  10
total node is found=5

1.Insert
3.Display
4.Search
0.Exit
enter your choice:4

which value search:
25

value position is=3
1.Insert
3.Display
4.Search
0.Exit
enter your choice:4

which value search:
50

 Data not found

1.Insert
```

```
2.DeLete
3.Display
4.Search
5.Sort

0.Exit
enter your choice:
```

6. ANALYSIS AND DISCUSSION

In first problem we get the proper use of link list insert data at three different position of the link list beginning,end,at any position of the link list and I understand how to use it link list of real life. In this problem I did not understand but finally I solved this problem. It was very useful in programming language made a list.