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: 04

Course Title: Data Structure Lab

Course Code: CSE 106 Section: DB

Lab Experiment Name: Linked List

Student Details

	Name	ID
1.	Shariful Islam Emon	213902056

Lab Date : 16/08/2022

Submission Date : 18/08/2022

Course Teacher's Name : Farhana Akter Sunny

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… Signature:.....

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

1. TITLE OF THE LAB EXPERIMENT

" Linked List "

2. IMPLEMENTATION

Answer to the problem no: 1

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

Code:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
//Shariful Islam Emon 213902056
 struct Node
 {
 int data;
 struct Node *previous, *next;
 }*head = NULL;

 void insertAtBeginning(int value)
 {
 struct Node *newNode;
 newNode = (struct Node*)malloc(sizeof(struct Node));
 newNode -> data = value;
 newNode -> previous = NULL;
 newNode -> next = NULL;
 if(head == NULL)
 {
 newNode -> next = NULL;
 head = newNode;
 }
 else
 {
 newNode -> next = head;
 head = newNode;
 }
```

```
printf("\nInsertion success!!!");
}

void insertAtEnd(int value)
{
struct Node *newNode;
newNode = (struct Node*)malloc(sizeof(struct
Node)); newNode -> data = value;
newNode -> next = NULL;
if(head == NULL)
{
newNode -> previous = NULL;
head = newNode;
}
else
{
struct Node *temp = head;
while(temp -> next != NULL)
temp = temp -> next;
temp -> next = newNode;
newNode -> previous = temp;
}
printf("\nInsertion success!!!");
}

void deleteBeginning()
{
if(head == NULL)
printf("List is Empty!!! Deletion not
possible!!!"); else
{
struct Node *temp = head;
if(temp -> previous == temp -> next)
{
head = NULL;
free(temp);
}
else{
head = temp -> next;
head -> previous = NULL;
free(temp);
}
printf("\nDeletion success!!!");
}
}

void display()
{
```

```
if(head == NULL)
printf("\nList is Empty!!!");
else
{
struct Node *temp = head;
printf("\nList elements are: \n");
printf("NULL <--- ");
while(temp -> next != NULL)
{
printf("%d <===> ",temp -> data);
temp = temp->next;
}
printf("%d ---> NULL", temp -> data);
}
}

int main()
{
int choice1, choice2, value;
while(1)
{
Start:
printf("\n*********** MENU *************\n");
printf("1. Insert\n2. Delete\n3. Display\n4. Exit\nEnter your choice: ")

;
scanf("%d",&choice1);
switch(choice1)
{
case 1: printf("Enter the value to be inserted: ");
scanf("%d",&value);
while(1)
{
printf("\nSelect from the following Inserting options\n"); printf("1.
At Beginning\n2. At End\n3. Cancel\nEnter your choice: ");
scanf("%d",&choice2);
switch(choice2)
{
case 1:
insertAtBeginning(value);
break;
case 2:
insertAtEnd(value);
break;
case 3:
goto EndSwitch;
default:
printf("\nPlease select correct Inserting
```

```
option!!!\n"); }
goto Start;
}
break;
case 2:
while(1)
{
printf("\nSelect from the following Deleting options\n");
printf("1. At Beginning\n2. Cancel\nEnter your choice:
"); scanf("%d",&choice2);
switch(choice2)
{
case 1:
deleteBeginning();
break;
case 2:
goto EndSwitch;
default:
printf("\nPlease select correct Deleting option!!!\n");
}
goto Start;
}
break;
EndSwitch:
break;
case 3:
display();
break;
case 4:
exit(0);
break;
default:
printf("\nPlease select correct option!!!");
}
}
return 0;
}
```

Output:

11:58 54%

Compile Result

```
**** MENU ****
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to be inserted: 8

Select from the following Inserting options
1. At Beginning
2. At End
3. Cancel
Enter your choice: 1

Insertion success!!!
**** MENU ****
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to be inserted: 7
```

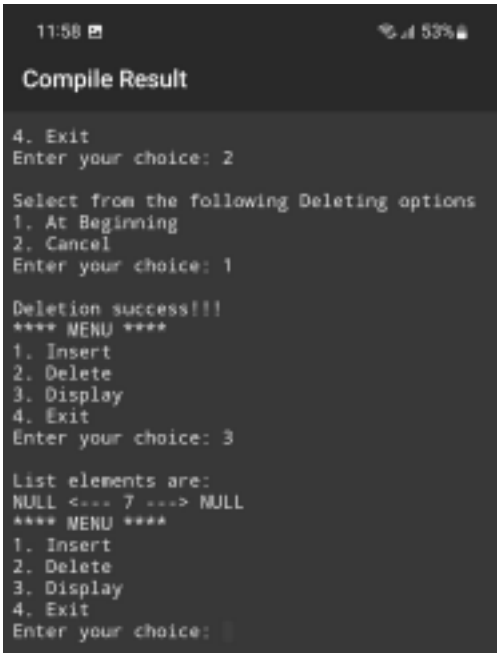
11:58 54%

Compile Result

```
Select from the following Inserting options
1. At Beginning
2. At End
3. Cancel
Enter your choice: 2

Insertion success!!!
**** MENU ****
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3

List elements are:
NULL <--- 8 <===> 7 ---> NULL
**** MENU ****
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
```



<u>Answer to the problem no: 2</u>

Problem Statement: Find the specific node of element that is present or not in the singly linked list

Code:

```
#include<stdio.h>
#include<stdlib.h>

struct node
{
    int data;
    struct node *next;
};

void addLast(struct node **head, int val)
{
    //create a new node
    struct node *newNode = malloc(sizeof(struct node));
    newNode->data = val;
    newNode->next = NULL;

    //if head is NULL, it is an empty list
    if(*head == NULL)
        *head = newNode;
    //Otherwise, find the last node and add the newNode
    else
    {
       struct node *lastNode = *head;
```

```
        //last node's next address will be NULL.
        while(lastNode->next != NULL)
        {
            lastNode = lastNode->next;
        }

        //add the newNode at the end of the linked list
        lastNode->next = newNode;
    }

}

int searchNode(struct node *head,int key)
{
    struct node *temp = head;

    //iterate the entire linked list and print the data
```

```
    while(temp != NULL)
    {
        //key found return 1.
        if(temp->data == key)
            return 1;
        temp = temp->next;
    }
    //key not found
    return -1;
}

int main()
{
    struct node *head = NULL;

    addLast(&head,10);
    addLast(&head,20);
    addLast(&head,30);

    //change the key and try it yourself.
    if(searchNode(head,20) == 1)
    {
        printf("Search Found");
    }

    else
    {
        printf("Search Not Found");
    }

    return 0;
}
```

Output: