Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: Summer 2022, B.Sc. in CSE (DAY)

LAB REPORT NO # 05

Course Title: Data Structure Lab
Course Code: CSE 106 **Section: CSE 213 - DA (PC)**

Lab Experiment Name(s):

- 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.

Student Details

Name	ID
Md. Shahidul Islam Prodhan	213902017

Lab Date: 10 August 2022

Submission Date: 17 August 2022

Course Teacher's Name: Ms Farhana Akter Sunny, Senior Lecturer.

[For Teacher's use only: Don't write anything inside this box]

Lab Report Status

Marks:	**Signature:**
Comments:	**Date:**

1. TITLE OF THE LAB EXPERIMENT

Write a C program to insert element at beginning, last and any specific position using linked list.

2. OBJECTIVES

Implement insertion of Link-list

3. PROCEDURE/ ANALYSIS / DESIGN

STEPS	Algorithm to insert an element from beginning
1	Initialize a Struct Node *newnode
2	Allocate memory Location for newnode
3	If (newnode == NULL) Print list is Full.
4	else Get newnode->data from user
5	set new->next = head
6	Set head = newnode
7	Exit

STEPS	Algorithm to insert an element from end
1	Initialize a Struct Node *newnode
2	Allocate memory Location for newnode
3	If (newnode == NULL) Print list is Full.
4	else Get newnode->data from user

5	if(head == NULL)
	newnode->next=NULL;
	head=newnode;

6	else
	temp=head;
	repeat while(temp->next!=NULL)
	temp = temp->next
	[end of if]

7	Set temp->next=newnode newnode->next=NULL
8	exit

3. PROCEDURE/ ANALYSIS / DESIGN

STEPS	Algorithm to insert an element at any position
1	Initialize a Struct Node *newnode
2	Allocate memory Location for newnode
3	Set i=1, and get the value of pos from user
4	If (newnode == NULL) Print list is Full.
5	else Get newnode->data from user

6	if(head == NULL)
	newnode->next=NULL;
	head=newnode;

7	elserepeat while(temp->next!=NULL) temp = temp->next set i=i+1
8	Set newnode->next=temp->next; temp->next=newnode;
9	exit

4. IMPLEMENTATION & TEST RESULT

Beginning insert

```
void beginsert(){
 struct node *newnode;
 int item,n;
 newnode=(struct node*)malloc(sizeof(struct node*));
  if(newnode==NULL){
     printf("List is full");
   }else{
   printf("enter the data:");
   scanf("%d",&newnode->data);
   //newnode->data;
   newnode->next=head;
   head=newnode;
   printf("\nNode Inserted\n");
   }

}
```

End Insert

```
void lastinsert(){
 struct node *newnode;
 int item;
 newnode=(struct node*)malloc(sizeof(struct node*));/
  if(newnode==NULL){
     printf("List is full");
   }else{
   printf("enter the data:");
   scanf("%d",&item);
   newnode->data=item;

     if(head==NULL){
     newnode->next=NULL;
     head=newnode;
     }else{
     temp=head;
      while(temp->next!=NULL)
      temp = temp->next;

     temp->next=newnode;
     newnode->next=NULL;
     printf("Inserted");
     }
   }
}
```

4. IMPLEMENTATION & TEST RESULT

Insert At Any Position

```
void insertaianyposition(){
struct node *newnode;
newnode=(struct node*)malloc(sizeof(struct node*));
int pos,i=1;
printf("\nEnter the position you want to Add:");
scanf("%d",&pos);

printf("\nEnter the data:");
scanf("%d",&newnode->data);
if(head==NULL){
    head=newnode;
    newnode->next =NULL;
    printf("\nNode Inserted At 1st Position.");
}else{
while(i<pos-1){
    temp=temp->next;
    i++;
}
newnode->next=temp->next;
temp->next=newnode;
printf("\nNode Inserted At %dth Position.",pos);
}
}
```

4. IMPLEMENTATION & TEST RESULT

```
1.Insert from Beg
2.Insert from end
3.Delete from Beg
4.Delete a specific elements
5.Search elements
6.Display
7.Delete from end
8.Sort The Elements
9.Insert at any Position
Enter your choice
1
enter the data:34

Node Inserted
```

```
1.Insert from Beg
2.Insert from end
3.Delete from Beg
4.Delete a specific elements
5.Search elements
6.Display
7.Delete from end
8.Sort The Elements
9.Insert at any Position
Enter your choice
2
enter the data:45
Inserted
```

```
Node Inserted

1.Insert from Beg
2.Insert from end
3.Delete from Beg
4.Delete a specific elements
5.Search elements
6.Display
7.Delete from end
8.Sort The Elements
9.Insert at any Position
Enter your choice
9

Enter the position you want to Add:2

Enter the data:10

Node Inserted At 2th Position.
```

```
1.Insert from Beg
2.Insert from end
3.Delete from Beg
4.Delete a specific elements
5.Search elements
6.Display
7.Delete from end
8.Sort The Elements
9.Insert at any Position
Enter your choice
6

elements are: 67
elements are: 34
elements are: 10
elements are: 45
There are 4 elements
```

1. TITLE OF THE LAB EXPERIMENT

Write a C program to Find the specific node of element that is present or not in the singly linked list.

2. OBJECTIVES

Search Elements in Linklist

3. PROCEDURE/ ANALYSIS / DESIGN

STEPS	Algorithm to insert an element from beginning
1	Get the findding data x from the user
2	Set temp = head
3	Repeat while (temp!=NULL)
4	if(temp->data == x) Print Data Founded break;
5	Set temp= temp->next
6	Exit

4. IMPLEMENTATION & TEST RESULT

Search

```
void search(){
//struct node *temp;
int x,c=0;
printf("\nenter the data you want to found:");
scanf("%d",&x);
temp=head;
while(temp!=NULL){
    if(temp->data == x){
        c++;
        break;
    }
    temp=temp->next;
}
if(c!=0){
    printf("founded\n");
}else{
printf("not found\n");
}
}
```

```
There are 4 elements
1.Insert from Beg
2.Insert from end
3.Delete from Beg
4.Delete a specific elements
5.Search elements
6.Display
7.Delete from end
8.Sort The Elements
9.Insert at any Position
Enter your choice
5

enter the data you want to found:10
founded
```

6. ANALYSIS AND DISCUSSION

1) This problem is solved by using c program. In this program we implement insertion of linklist.
2) This problem is solved by using c program. In this program we implement Searching of
3) linklist.

7. SUMMARY

1. We done this problem in c programming language.