Experiment - 1.3

Student Name: Aayush Tyagi **UID: 21BCS7312**
Branch: CSE **Section/Group: 717-A**
Semester: 3rd **Date of Performance:09/08/22**
Subject Name: Data Structures **Subject Code: 21CSH211**

1. **Aim of the practical:**
   Write a menu driven program that maintains a linear linked list whose elements are stored in on ascending order and implements the following operations (using separate functions):
   a) Insert a new element
   b) Delete an existing element
   c) Search an element
   d) Display all the elements

2. **Algorithm:**
   Insertion
   a) Step 1. Start
   b) Step 2. Initialize the node.
   c) Step 3. Take a constant temp variable=(struct node)
   d) Step 4. If (temp==null)
      Full;
   e) Step 5. Input the value and position to insert.
   f) Step 6. If (pos==0) found at start
      Else
      Use for loop until (ptr==null)
      Insert value.
   g) Step 7. End

Display

a) Step 1. Start
b) Step 2. Store data in nodes.
c) Step 3. If(start==Null)
List is full;
Else
Ptr=start;
Use while loop till (ptr!=null)
d) Step 4. Display if condition is true.
e) Step 5. Exit

Deletion

a) Step 1. Start
b) Step 2. Store data in nodes.
c) Step 3. If(Start==Null)
List is empty.
Exit.
d) Step 4. If(pos==0)
Ptr=start;
Start=start->next;
Else
If(ptr==Null)
Go to step 5
e) Step 5. Exit

3. **Program Code:**

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

void create();
void display();
void insert_pos();
void delete_pos();


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

struct node *start=NULL;
int main()
{
        int choice;
        while(1)
        {
            printf("1.Create \n");
            printf("2.Display \n");
            printf("3.Insert at specified position \n");
            printf("4.Delete from specified position \n");
            printf("5.Exit \n");
            printf("Enter your choice: ");
            scanf("%d",&choice);
            switch(choice)
            {
                case 1:
                                create();
                                break;
                case 2:
                                display();
                                printf("Displayed List \n ");
                                exit(0);
```

```
                        break;
                case 3:
                        insert_pos();
                        break;
                case 4:
                        delete_pos();
                        break;
                case 5:
                        printf("Exited");
                        exit(0);
                        break;
                default:
                        printf("Invalid Choice \n");
                        break;
                }
        }
        return 0;
}
void create()
{
        struct node *temp,*ptr;
        temp=(struct node *)malloc(sizeof(struct node));
        if(temp==NULL)
        {
                printf("Full : ");
                exit(0);
        }
        printf("Enter the data value for the node : ");
        scanf("%d",&temp->info);
        temp->next=NULL;
        if(start==NULL)
        {
                start=temp;
        }
        else
        {
            ptr=start;
```

```
            while(ptr->next!=NULL)
            {
                ptr=ptr->next;
            }
            ptr->next=temp;
        }
}
void display()
{
        struct node *ptr;
        if(start==NULL)
        {
            printf("List is empty \n");
            return;
        }
        else
        {
            ptr=start;
            printf("The List elements are : ");
            while(ptr!=NULL)
            {
                printf("%d\n",ptr->info );

                ptr=ptr->next ;
            }
        }
}

void insert_pos()
{
        struct node *ptr,*temp;
        int i,pos;
        temp=(struct node *)malloc(sizeof(struct node));
        if(temp==NULL)
        {
            printf("Full\n");
            return;
```

```
        }
        printf("Enter the position for the new node to be inserted : ");
        scanf("%d",&pos);
        printf("Enter the data value of the node : ");
        scanf("%d",&temp->info) ;

        temp->next=NULL;
        if(pos==0)
        {
                temp->next=start;
                start=temp;
        }
        else
        {
                for(i=0,ptr=start;i<pos-1;i++) { ptr=ptr->next;
                        if(ptr==NULL)
                        {
                                printf(" Position not found \n");
                                return;
                        }
                }
                temp->next =ptr->next ;
                ptr->next=temp;
        }
}
void delete_pos()
{
        int i,pos;
        struct node *temp,*ptr;
        if(start==NULL)
        {
                printf(" The List is Empty: ");
                exit(0);
        }
        else
        {
                printf(" Enter the position of the node to be deleted: ");
```

```
                scanf("%d",&pos);
                if(pos==0)
                {
                        ptr=start;
                        start=start->next ;
                        printf(" The deleted element is : %d \n",ptr->info  );
                        free(ptr);
                }
                else
                {
                        ptr=start;
                        for(i=0;i<pos;i++) { temp=ptr; ptr=ptr->next ;
                                if(ptr==NULL)
                                {
                                        printf(" Position not Found \n ");
                                        return;
                                }
                        }
                        temp->next =ptr->next ;
                        printf(" The deleted element is : %d ",ptr->info );
                        free(ptr);
                }
        }
}
```

4. Output:

```
Enter the data value for the node : 30
1.Create
2.Display
3.Insert at specified position
4.Delete from specified position
5.Exit
Enter your choice: 3
Enter the position for the new node to be inserted : 1
Enter the data value of the node : 25
1.Create
2.Display
3.Insert at specified position
4.Delete from specified position
5.Exit
Enter your choice: 2
The List elements are : 10
25
20
30
Displayed List

PS D:\Data Structures\.vscode>
```

5. **Learning outcomes (What I have learnt):**

1. Learnt about searching in linked list.
2. Learnt about insertion in linked list.
3. Learnt about different operation on a linked list**.**
4. Learnt how to create a node in linked list.