LAB ASSESSMENT 2

NAME: K. TRIVIKRAM SAI

REGNO: 22BCE3280

MENU DRIVEN PROGRAM ON LINKED LIST

```c
#include<stdio.h>
#include<stdlib.h>
int count=0;
struct Node *start=NULL;
struct Node
{
    int data;
    struct Node *next;
};
void insert_at_begin(int x)
{
    struct Node *t;
    t=(struct Node*)malloc(sizeof(struct Node));
    count++;
    if(start==NULL)
    {
        start=t;
        start->data=x;
        start->next=NULL;
        return;
    }
    t->data=x;
    t->next=start;
    start=t;
}
void insertAtPosition()
{
    struct Node *temp, *newnode;
    int pos, dat, i = 1;
    newnode = malloc(sizeof(struct Node));
    printf("\nEnter position and data :");
    scanf("%d %d", &pos, &dat);

    temp = start;
    newnode->data = dat;
```

```
    newnode->next = 0;
    while (i < pos - 1) {
        temp = temp->next;
        i++;
    }
    newnode->next = temp->next;
    temp->next = newnode;
}
void insert_at_end(int x)
{
    struct Node *t,*temp;
    t=(struct Node*)malloc(sizeof(struct Node));
    count++;
    if(start==NULL)
    {
        start=t;
        start->data=x;
        start->next=NULL;
        return;
    }
    temp=start;
    while(temp->next!=NULL)
    {
        temp=temp->next;
    }
    temp->next=t;
    t->data=x;
    t->next=NULL;
}
void delete_from_begin()
{
    struct Node *t;
    int n;
    if(start==NULL)
    {
        printf("Linked List is empty!!!\n\n");
        return;
    }
    n=start->data;
    t=start->next;
    free(start);
    start=t;
    count--;
    printf("Deleted element is %d\n\n",n);
    return;
}
void delete_from_position()
{
```

```
    struct Node *temp, *position;
    int i = 1, pos;
    if (start == NULL)
        printf("\nList is empty\n");
    else {
        printf("\nEnter index : ");
        scanf("%d", &pos);
        position = malloc(sizeof(struct Node));
        temp = start;
        while (i < pos - 1) {
            temp = temp->next;
            i++;
        }
        position = temp->next;
        temp->next = position->next;
    }
}

void delete_from_end()
{
    struct Node *t,*u;
    int n;
    if(start==NULL)
    {
        printf("Linked List is empty!!!\n\n");
        return;
    }
    count--;
    if(start->next==NULL)
    {
        n=start->data;
        free(start);
        start=NULL;
        printf("Deleted element is %d\n\n",n);
        return;
    }
    t=start;
    while(t->next!=NULL)
    {
        u=t;
        t=t->next;
    }
    n=t->data;
    u->next=NULL;
    free(t);
    printf("Deleted element is %d\n\n",n);
    return;
}
```

```c
void display()
{
    struct Node *t;
    if(start==NULL)
    {
        printf("Linked List is empty!!!\n\n");
        return;
    }
    printf("No of elements: %d\n",count);
    printf("Elements are: ");
    t=start;
    while(t->next!=NULL)
    {
        printf("%d ",t->data);
        t=t->next;
    }
    printf("%d ",t->data);
    printf("\n\n");
}
void search()
{
    int found = -1;
    struct Node* tr = start;
    if (start == NULL) {
        printf("Linked list is empty\n");
    }
    else {
        printf("\nEnter the element you want to search: ");
        int key;
        scanf("%d", &key);
        while (tr != NULL) {
            if (tr->data == key) {
                found = 1;
                break;
            }
            else {
                tr = tr->next;
            }
        }
        if (found == 1) {
            printf("Yes, %d is present in the linked list.\n",key);
        }
        else {
            printf("No, %d is not present in the linked list.\n",key);
        }
    }
}
```

```c
int main()
{
    int ch,data;
    while(1)
    {
        printf("---LINKED LIST PROGRAMS---\n");
        printf("1. INSERT AT BEGINING\n");
        printf("2. INSERT AT END\n");
        printf("3. INSERT AT POSITION\n");
        printf("4. DELETE FROM BEGINING\n");
        printf("5. DELETE FROM END\n");
        printf("6. DELETE AT POSITION\n");
        printf("7. DISPLAY LIST\n");
        printf("8. SEARCH FOR ELEMENT IN LIST\n");
        printf("9. EXIT\n\n");
        printf("Enter your choice: ");
        scanf("%d",&ch);
        if(ch==1)
        {
            printf("Enter the insert value: ");
            scanf("%d",&data);
            printf("\n");
            insert_at_begin(data);
        }
        else if(ch==2)
        {
            printf("Enter the insert value: ");
            scanf("%d",&data);
            printf("\n");
            insert_at_end(data);
        }
        else if(ch==3)
        {
            insertAtPosition();
        }
        else if(ch==4)
        {
            delete_from_begin();
        }
        else if(ch==5)
        {
            delete_from_end();
        }
        else if(ch==6)
        {
            delete_from_position();
        }
        else if(ch==7)
```

```
        {
            display();
        }
        else if(ch==8)
        {
            search();
        }
        else if(ch==9)
        {
            break;
        }
        else
        {
            printf("Wrong choice!!!\n");
        }
    }
}
```

OUTPUT

```
---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT

Enter your choice: 1
Enter the insert value: 1

---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT

Enter your choice: 1
Enter the insert value: 5

---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT
```

```
Enter your choice: 2
Enter the insert value: 6

---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT

Enter your choice: 3

Enter position and data :2
8
---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT

Enter your choice: 7
No of elements: 3
Elements are: 5 8 1 6
```

```
Enter your choice: 4
Deleted element is 5

---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT

Enter your choice: 5
Deleted element is 6

---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT

Enter your choice: 6

Enter index : 1
```

```
---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT

Enter your choice: 7
No of elements: 1
Elements are: 8

---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT

Enter your choice: 2
Enter the insert value: 6

---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT
```

```
Enter your choice: 7
No of elements: 2
Elements are: 8 6

---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT

Enter your choice: 8

Enter the element you want to search: 6
Yes, 6 is present in the linked list.
---LINKED LIST PROGRAMS---
1. INSERT AT BEGINING
2. INSERT AT END
3. INSERT AT POSITION
4. DELETE FROM BEGINING
5. DELETE FROM END
6. DELETE AT POSITION
7. DISPLAY LIST
8. SEARCH FOR ELEMENT IN LIST
9. EXIT

Enter your choice: 9


------------------
(program exited with code: 0)

Press any key to continue . . .
```