Shrihari K Mangle
CSB2-74 12010517

LAB-3

Implement SLL for the following operations:

1. Create
2. Display
3. Insert.
4. Delete.
5. Reverse.
6. Concatenate

CODE :

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

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

void printLL(node *head)
{
    while (head != NULL)
    {
        printf("%d -> ", head->data);
        head = head->next;
    }
    printf("NULL");
    printf("\n");
}

// CREATING A LIST

node *createNode(int value)
{
    node *p = (node *)malloc(sizeof(node));
```

```
    p->data = value;
    p->next = NULL;

    return p;
}

// INSERTING OPERATIONS

node *insertAtBeg(node *head, int value)
{
    node *p = createNode(value);

    if (p == NULL)
    {
        return p;
    }

    else
    {
        p->next = head;
        head = p;
    }

    return head;
}

node *insertAtloc(node *head, int position, int value)
{
    node *p = createNode(value);
    node *q = head;

    if (position == 1)
    {
        head = insertAtBeg(head, value);
        return head;
    }

    for (int i = 1; i < position - 1; i++)
    {
        q = q->next;
    }

    p->next = q->next;
    q->next = p;

    return head;
}

node *insertAtEnd(node *head, int value)
{
    node *p = createNode(value);
```

```
    node *q = head;

    if (q == NULL)
    {
        return p;
    }

    while (q->next != NULL)
    {
        q = q->next;
    }

    q->next = p;

    return head;
}

// DELETING FUNCTIONS

node *DeleteAtBeg(node *head)
{
    node *p = head;
    if (head == NULL)
    {
        printf("Cannot delete from empty list\n");
        return NULL;
    }
    head = head->next;
    free(p);

    return head;
}

node *DeleteAtIndex(node *head, int index)
{
    node *p = head;
    node *q = head->next;

    if (index == 1)
    {
        head = DeleteAtBeg(head);
        return head;
    }

    int i = 1;
    while (i != index - 1)
    {
        p = p->next;
        q = q->next;
        i++;
    }
```

```
    p->next = q->next;
    free(q);

    return head;
}

node *DeleteAtEnd(node *head)
{

    node *p = head;
    node *q = head->next;
    if (p->next == NULL)
    {
        free(p);
        return NULL;
    }
    while (q->next != NULL)
    {
        p = p->next;
        q = q->next;
    }

    p->next = NULL;
    free(q);

    return head;
}

// REVERSING FUNCTION

node *reverseLL(node *head)
{
    node *p, *q, *r;
    p = NULL;
    q = r = head;

    while (q != NULL)
    {
        r = r->next;
        q->next = p;
        p = q;
        q = r;
    }

    head = p;

    return head;
}

// CONCATINATING THE EXISTING WITH THE ANOTHER ONE
```

Shrihari K Mangle
CSB2-74 12010517

```c
node *concatLL(node *head1, node *head2)
{

    if (head1 == NULL)
    {
        return head2;
    }
    else if (head2 == NULL)
    {
        return head1;
    }

    else
    {
        node *p = head1;
        while (p->next != NULL)
        {
            p = p->next;
        }
        p->next = head2;

        return head1;
    }
}

// MAIN FUNCTION

int main()
{

    int choice;
    int x = 0;
    node *head = NULL;

    do
    {
        printf("CHOICE : \n 1.CREATE LINKED-LIST\n 2.INSERT AT BEGINNING\n
3.INSERT AT PARTICULAR LOCATION\n 4.INSERT AT END\n 5.DISPLAY\n 6.DELETE
FROM BEGINNING\n 7.DELETE AT PARTICULAR INDEX\n 8.DELETE AT END\n 9.REVERSE
LL\n 10.CONCAT LL\n");
        printf("\nEnter choice : ");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            printf("\nEnter the no. of nodes u want to create : ");
            int nodes, values;
            scanf("%d", &nodes);
            printf("\n");
```

```
            for (int i = 0; i < nodes; i++)
            {
                printf("Enter value : ");
                scanf("%d", &values);
                head = insertAtEnd(head, values);
            }
            printf("\nLinked list created successfully\n");

            break;

        case 2:
            printf("Enter the element to insert : ");
            int element;
            scanf("%d", &element);
            head = insertAtBeg(head, element);
            printf("Element added successfully\n");
            break;

        case 3:
            printf("Enter the element to insert : ");
            int e, location;
            scanf("%d", &e);
            printf("Enter the location : ");
            scanf("%d", &location);
            head = insertAtloc(head, location, e);
            printf("Element added successfully\n");
            break;

        case 4:
            printf("Enter the element to insert : ");
            int ele;
            scanf("%d", &ele);
            head = insertAtEnd(head, ele);
            printf("Element added successfully\n");

            break;

        case 5:
            printLL(head);
            break;

        case 6:
            head = DeleteAtBeg(head);
            printf("Element deleted successfully\n");
            break;

        case 7:
            printf("Enter the index : ");
            int index;
            scanf("%d", &index);
            head = DeleteAtIndex(head, index);
```

```
            printf("Element deleted successfully\n");
            break;

        case 8:
            head = DeleteAtEnd(head);
            printf("Element deleted successfully\n");
            break;

        case 9:
            head = reverseLL(head);
            printf("List is reversed successfully\n");
            break;

        case 10:
            printf("\nEnter the no. of nodes u want to create in second LL
: ");
            node *headx = NULL;
            int n, v;
            scanf("%d", &n);
            printf("\n");
            for (int i = 0; i < n; i++)
            {
                printf("Enter value : ");
                scanf("%d", &v);
                headx = insertAtEnd(headx, v);
            }
            printf("The two LL are : \n");
            printLL(head);
            printLL(headx);

            head = concatLL(head, headx);
            printf("The list after concatination is :\n");
            printLL(head);
            break;
        }
        printf("Do you want to continue 1/0 : ");
        scanf("%d", &x);

        printf("\n");
    } while (x != 0);

    return 0;
}
```

Shrihari K Mangle
CSB2-74 12010517

OUTPUT :

CASE 1 : CREATE THE LINKED-LIST

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\VIT\SECOND YEAR(SY)\SEM 2\DATA STRUCTURES(DS)\DATA STRUCTURES LAB>
AB\" ; if ($?) { gcc SinglyLL.c -o SinglyLL } ; if ($?) { .\SinglyLL }
CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 1

Enter the no. of nodes u want to create : 5

Enter value : 6
Enter value : 8
Enter value : 3
Enter value : 9
Enter value : 2

Linked list created is : 6 -> 8 -> 3 -> 9 -> 2 -> NULL

Do you want to continue 1/0 : 1
```

Shrihari K Mangle
CSB2-74 12010517

CASE 2 : INSERT AT BEGINNING

```
CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 2
Enter the element to insert : 1
Element added successfully
Do you want to continue 1/0 : 1

CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 5
1 -> 6 -> 8 -> 3 -> 9 -> 2 -> NULL
Do you want to continue 1/0 : 1
```

Shrihari K Mangle
CSB2-74 12010517

CASE 3: INSERT AT PARTICULAR LOCATION

```
CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 3
Enter the element to insert : 4
Enter the location : 3
Element added successfully
Do you want to continue 1/0 : 1

CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 5
1 -> 6 -> 4 -> 8 -> 3 -> 9 -> 2 -> NULL
Do you want to continue 1/0 : 1
```

Shrihari K Mangle
CSB2-74 12010517

CASE 4: INSERT AT END

```
CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 4
Enter the element to insert : 5
Element added successfully
Do you want to continue 1/0 : 1

CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 5
1 -> 6 -> 4 -> 8 -> 3 -> 9 -> 2 -> 5 -> NULL
Do you want to continue 1/0 : 1
```

Shrihari K Mangle
CSB2-74 12010517

CASE 6 : DELETE FROM BEGINNING

```
CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 6
Element deleted successfully
Do you want to continue 1/0 : 1

CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 5
6 -> 4 -> 8 -> 3 -> 9 -> 2 -> 5 -> NULL
Do you want to continue 1/0 : 1s
```

Shrihari K Mangle
CSB2-74 12010517

CASE 7 : DELETE AT PARTICULAR INDEX

```
CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 7
Enter the index : 5
Element deleted successfully
Do you want to continue 1/0 : 1

CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 5
6 -> 4 -> 8 -> 3 -> 2 -> 5 -> NULL
Do you want to continue 1/0 :
```

Shrihari K Mangle
CSB2-74 12010517

CASE 8 : DELETE AT END

```
CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 8
Element deleted successfully
Do you want to continue 1/0 : 1

CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 5
6 -> 4 -> 8 -> 3 -> 2 -> NULL
Do you want to continue 1/0 :
```

Shrihari K Mangle
CSB2-74 12010517

CASE 9 : REVERSING

```
CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 9
List is reversed successfully
Do you want to continue 1/0 : 1

CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 5
2 -> 3 -> 8 -> 4 -> 6 -> NULL
Do you want to continue 1/0 :
```

Shrihari K Mangle
CSB2-74 12010517

CASE 10 : CONCATINATION

```
CHOICE :
 1.CREATE LINKED-LIST
 2.INSERT AT BEGINNING
 3.INSERT AT PARTICULAR LOCATION
 4.INSERT AT END
 5.DISPLAY
 6.DELETE FROM BEGINNING
 7.DELETE AT PARTICULAR INDEX
 8.DELETE AT END
 9.REVERSE LL
 10.CONCAT LL

Enter choice : 10

Enter the no. of nodes u want to create in second LL : 4

Enter value : 7
Enter value : 4
Enter value : 1
Enter value : 2
The two LL are :
2 -> 3 -> 8 -> 4 -> 6 -> NULL
7 -> 4 -> 1 -> 2 -> NULL
The list after concatination is :
2 -> 3 -> 8 -> 4 -> 6 -> 7 -> 4 -> 1 -> 2 -> NULL
Do you want to continue 1/0 : 0

PS E:\VIT\SECOND YEAR(SY)\SEM 2\DATA STRUCTURES(DS)\DATA STRUCTURES LAB>
```