LAB 6

Name : Abhinav Sanjay

USN : 1BM23CS009

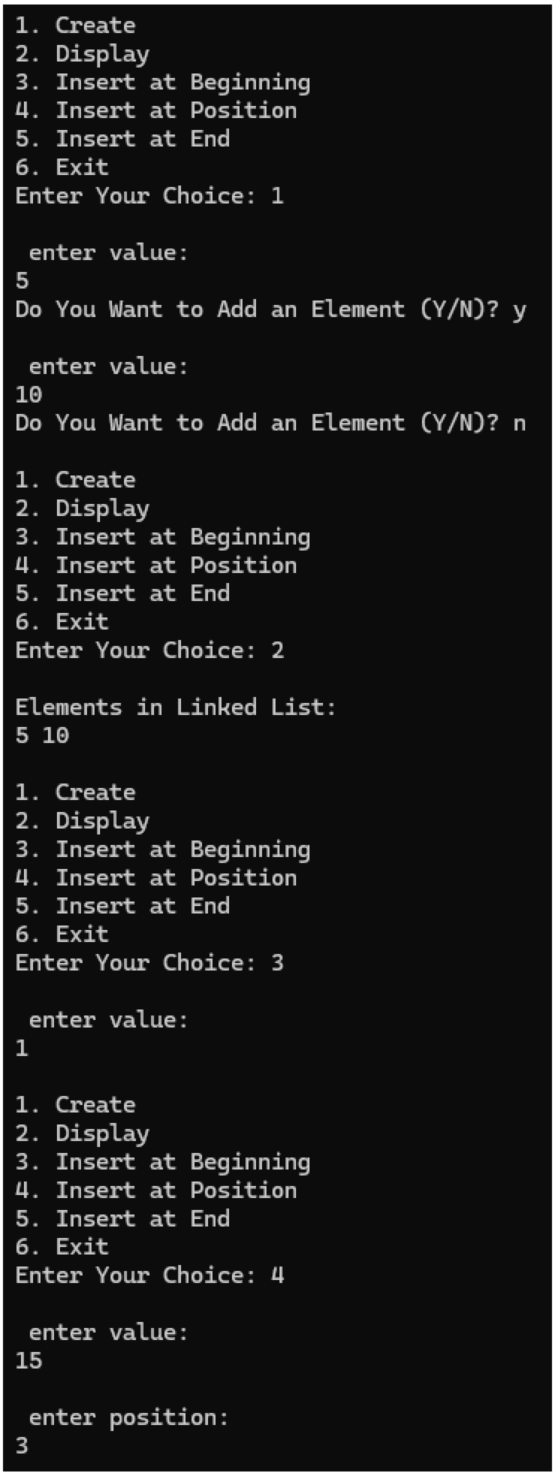
1) WAP to Implement Singly Linked List with following operations

a) Create a linked list.
b) Insertion of a node at first position, at any position and at end of list.
c) Display the contents of the linked list.

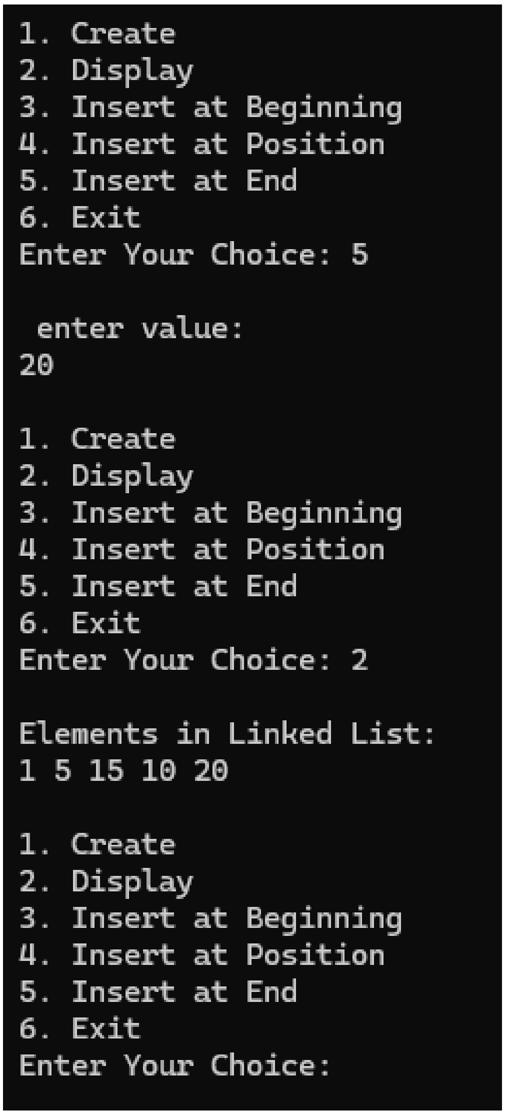
```
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int data;          // Data for the node
    struct Node *link;  // Pointer to the next node in the list
};

typedef struct Node node;
node *start = NULL;  // Start of the linked list, initially NULL
node *new1, *curr, *ptr; // Global declaration for new1, curr, and ptr

// Function prototypes
void create();
void display();
void InsertStart();
void InsertPosition();
void InsertEnd();
void main() {
    int ch;
    while (1) {
        printf("\n1. Create \n2. Display \n3. Insert at Beginning \n4. Insert at Position \n5. Insert at End \n6. Exit");
        printf("\nEnter Your Choice: ");
        scanf("%d", &ch);

        switch (ch) {
            case 1:
                create();
                break;
            case 2:
                display();
                break;
            case 3:
                InsertStart();
                break;
            case 4:
                InsertPosition();
                break;
            case 5:
                InsertEnd();
                break;
            case 6:
                exit(0);
            default:
                printf("Enter a Number between 1 and 6.\n");
        }
    }
}

void create() {
    char ch;

    do {
        new1 = (node*)malloc(sizeof(node));
       printf("\n enter value:\n");
       scanf("%d",&new1->data);
       if (start==NULL)
       {
           start=new1;
           curr=new1;
       }
       else {
           curr->link = new1;
           curr=new1;
       }

        printf("Do You Want to Add an Element (Y/N)? ");
        scanf(" %c", &ch);
    } while (ch == 'y' || ch == 'Y');
    curr->link=NULL;
}
```

```
void InsertStart() {
    new1 = (node*)malloc(sizeof(node));
    printf("\n enter value:\n");
    scanf("%d",&new1->data);
    if(start==NULL)
    {
        start=new1;
        new1->link=NULL;
        return;
    }
    else {
        new1->link=start;
        start=new1;
        return;
    }
}

void InsertPosition() {
    new1 = (node*)malloc(sizeof(node));
    printf("\n enter value:\n");
    scanf("%d",&new1->data);
    if(start==NULL)
    {
        start=new1;
        new1->link=NULL;
        return;
    }
    int i=1, pos;
    ptr=start;
    printf("\n enter position:\n");
    scanf("%d",&pos);
    while (ptr!=NULL && i<pos-1)
    {
        ptr=ptr->link;
        i++;
    }
    if(ptr==NULL)
    {
        return;
    }

    new1->link=ptr->link;
    ptr->link=new1;
}

void InsertEnd() {
    new1 = (node*)malloc(sizeof(node));
    printf("\n enter value:\n");
    scanf("%d",&new1->data);
    if(start==NULL)
    {
        start=new1;
        new1->link=NULL;
        return;
    }

    ptr=start;
    while(ptr->link !=NULL)
    {
        ptr=ptr->link;
    }
    ptr->link=new1;
    new1->link=NULL;
    return;
}

void display() {
    if (start == NULL) {
        printf("\nLinked List is Empty.\n");
        return;
    }
    ptr = start;
    printf("\nElements in Linked List: \n");

    while (ptr != NULL) {
        printf("%d ", ptr->data);
        ptr = ptr->link;
    }
    printf("\n");
}
```

Output



```
1. Create
2. Display
3. Insert at Beginning
4. Insert at Position
5. Insert at End
6. Exit
Enter Your Choice: 1

 enter value:
5
Do You Want to Add an Element (Y/N)? y

 enter value:
10
Do You Want to Add an Element (Y/N)? n

1. Create
2. Display
3. Insert at Beginning
4. Insert at Position
5. Insert at End
6. Exit
Enter Your Choice: 2

Elements in Linked List:
5 10

1. Create
2. Display
3. Insert at Beginning
4. Insert at Position
5. Insert at End
6. Exit
Enter Your Choice: 3

 enter value:
1

1. Create
2. Display
3. Insert at Beginning
4. Insert at Position
5. Insert at End
6. Exit
Enter Your Choice: 4

 enter value:
15

 enter position:
3
```



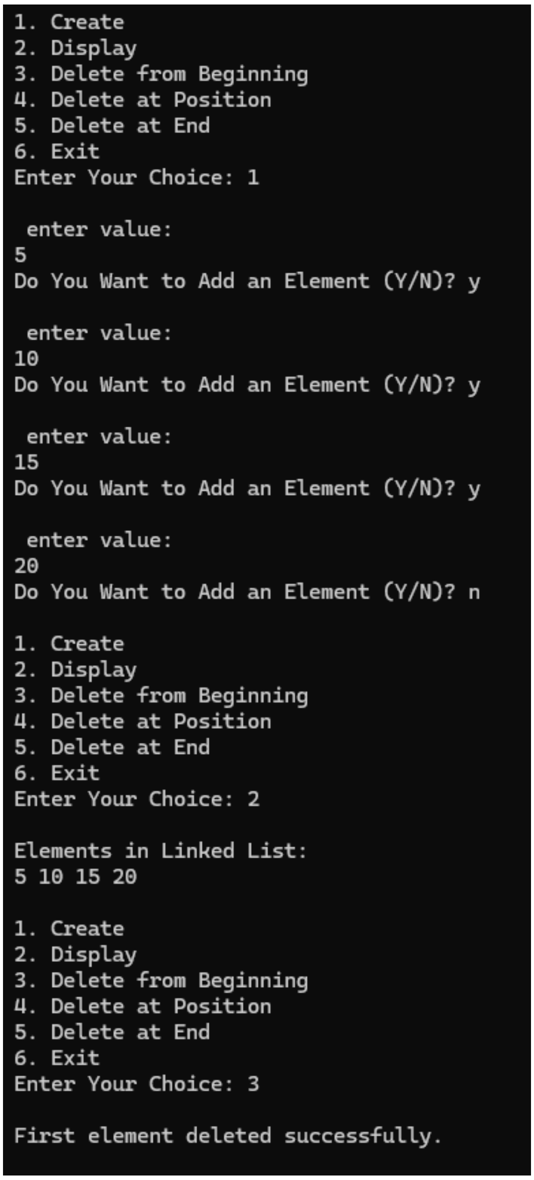
```
1. Create
2. Display
3. Insert at Beginning
4. Insert at Position
5. Insert at End
6. Exit
Enter Your Choice: 5

 enter value:
20

1. Create
2. Display
3. Insert at Beginning
4. Insert at Position
5. Insert at End
6. Exit
Enter Your Choice: 2

Elements in Linked List:
1 5 15 10 20

1. Create
2. Display
3. Insert at Beginning
4. Insert at Position
5. Insert at End
6. Exit
Enter Your Choice:
```

2) WAP to Implement Singly Linked List with following operations

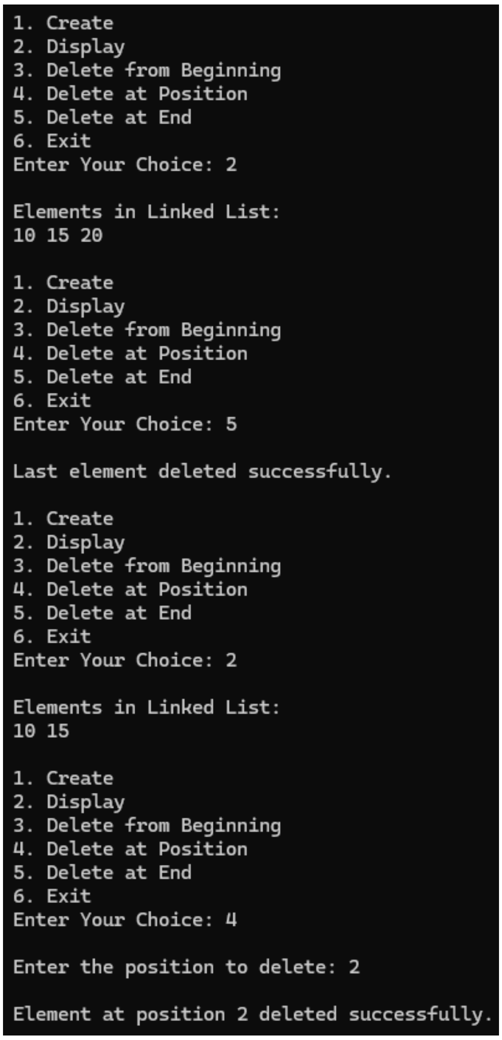
a) Create a linked list.
b) Deletion of first element, specified element and last element in the list.
c) Display the contents of the linked list.

```
#include <stdio.h>

#include <stdlib.h>


struct Node {

    int data;

    struct Node *link;

};


typedef struct Node node;

node *start = NULL;


void create();

void display();

void DeletefromStart();

void DeleteatPosition();

void DeleteatEnd();


void main() {

    int ch;

    while (1) {

        printf("\n1. Create \n2. Display \n3. Delete from Beginning \n4. Delete at 
Position \n5. Delete at End \n6. Exit");

        printf("\nEnter Your Choice: ");

        scanf("%d", &ch);
        switch (ch) {
            case 1:
                create();
                break;
            case 2:
                display();
                break;
            case 3:
                DeletefromStart();
                break;
            case 4:
                DeleteatPosition();
                break;
            case 5:
                DeleteatEnd();
                break;
            case 6:
                exit(0);
            default:
                printf("Enter a Number between 1 and 9.\n");
        }
    }
}

void create() {
    char ch;
    node *new1, *curr;
    do {
        new1 = (node*)malloc(sizeof(node));
       printf("\n enter value:\n");
       scanf("%d",&new1->data);
       if (start==NULL)
       {
           start=new1;
           curr=new1;
       }
       else {
           curr->link = new1;
           curr=new1;
       }

        printf("Do You Want to Add an Element (Y/N)? ");
        scanf(" %c", &ch);
    } while (ch == 'y' || ch == 'Y');
    curr->link=NULL;
}

void display() {
    if (start == NULL) {
        printf("\nLinked List is Empty.\n");
        return;
    }

    node *temp = start;
    printf("\nElements in Linked List: \n");
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->link;
    }
    printf("\n");
}

void DeletefromStart() {
    if (start == NULL) {
        printf("\nLinked List is Empty.\n");
        return;
    }

    node *temp = start;
    start = start->link;
    free(temp);
    printf("\nFirst element deleted successfully.\n");
}

void DeleteatPosition() {
    int pos, i = 1;
    if (start == NULL) {
        printf("\nLinked List is Empty.\n");
        return;
    }

    printf("\nEnter the position to delete: ");
    scanf("%d", &pos);
    node *temp = start;
    node *prev = NULL;

    if (pos == 1) {
        start = temp->link;
        free(temp);
        printf("\nElement at position %d deleted successfully.\n", pos);
        return;
    }

    while (temp != NULL && i < pos) {
        prev = temp;
        temp = temp->link;
        i++;
    }

    if (temp == NULL) {
        printf("\nPosition not found.\n");
        return;
    }

    prev->link = temp->link;
    free(temp);
    printf("\nElement at position %d deleted successfully.\n", pos);
}

void DeleteatEnd() {
    if (start == NULL) {
        printf("\nLinked List is Empty.\n");
        return;
    }

    node *temp = start;
    node *prev = NULL;

    if (start->link == NULL) {
        start = NULL;
        free(temp);
        printf("\nLast element deleted successfully.\n");
        return;
    }

    while (temp->link != NULL) {
        prev = temp;
        temp = temp->link;
    }

    prev->link = NULL;
    free(temp);
    printf("\nLast element deleted successfully.\n");
}
```

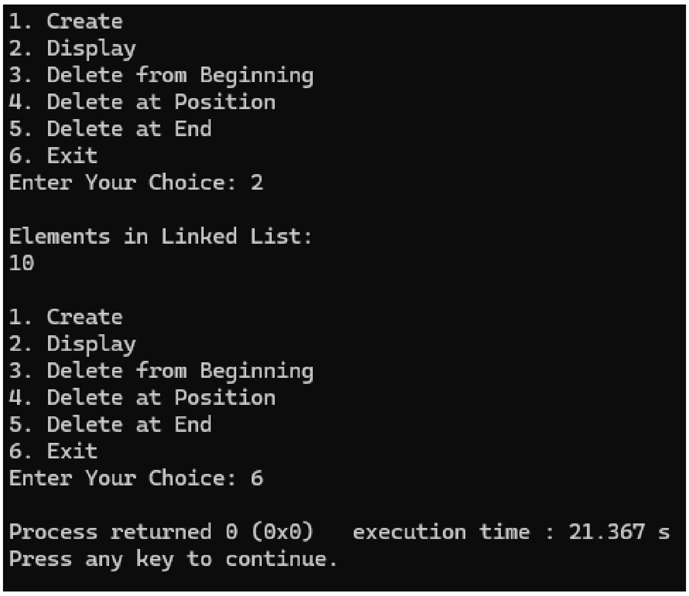
Output:



```
1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 1

 enter value:
5
Do You Want to Add an Element (Y/N)? y

 enter value:
10
Do You Want to Add an Element (Y/N)? y

 enter value:
15
Do You Want to Add an Element (Y/N)? y

 enter value:
20
Do You Want to Add an Element (Y/N)? n

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 2

Elements in Linked List:
5 10 15 20

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 3

First element deleted successfully.
```



```
1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 2

Elements in Linked List:
10 15 20

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 5

Last element deleted successfully.

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 2

Elements in Linked List:
10 15

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 4

Enter the position to delete: 2

Element at position 2 deleted successfully.
```



```
1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 2

Elements in Linked List:
10

1. Create
2. Display
3. Delete from Beginning
4. Delete at Position
5. Delete at End
6. Exit
Enter Your Choice: 6

Process returned 0 (0x0)   execution time : 21.367 s
Press any key to continue.
```