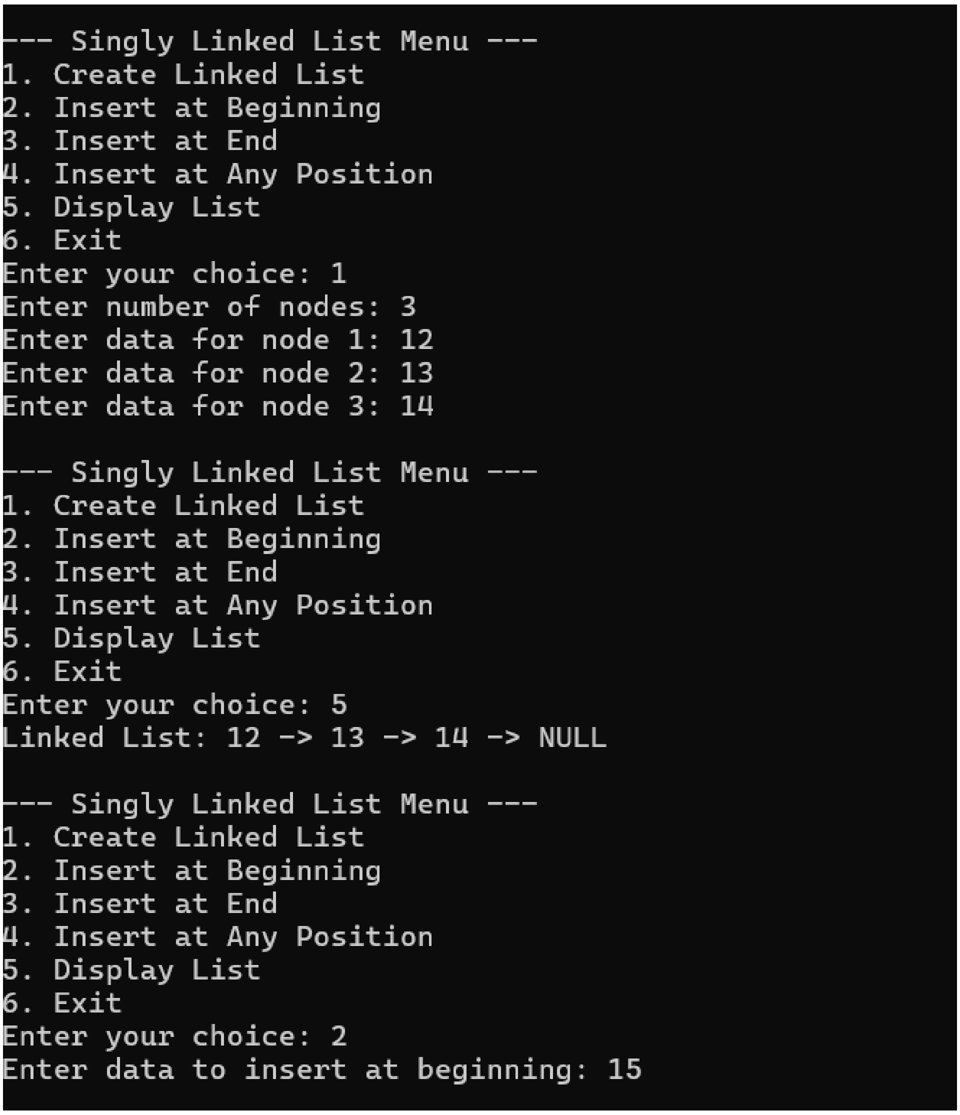
Lab pogram 4:-

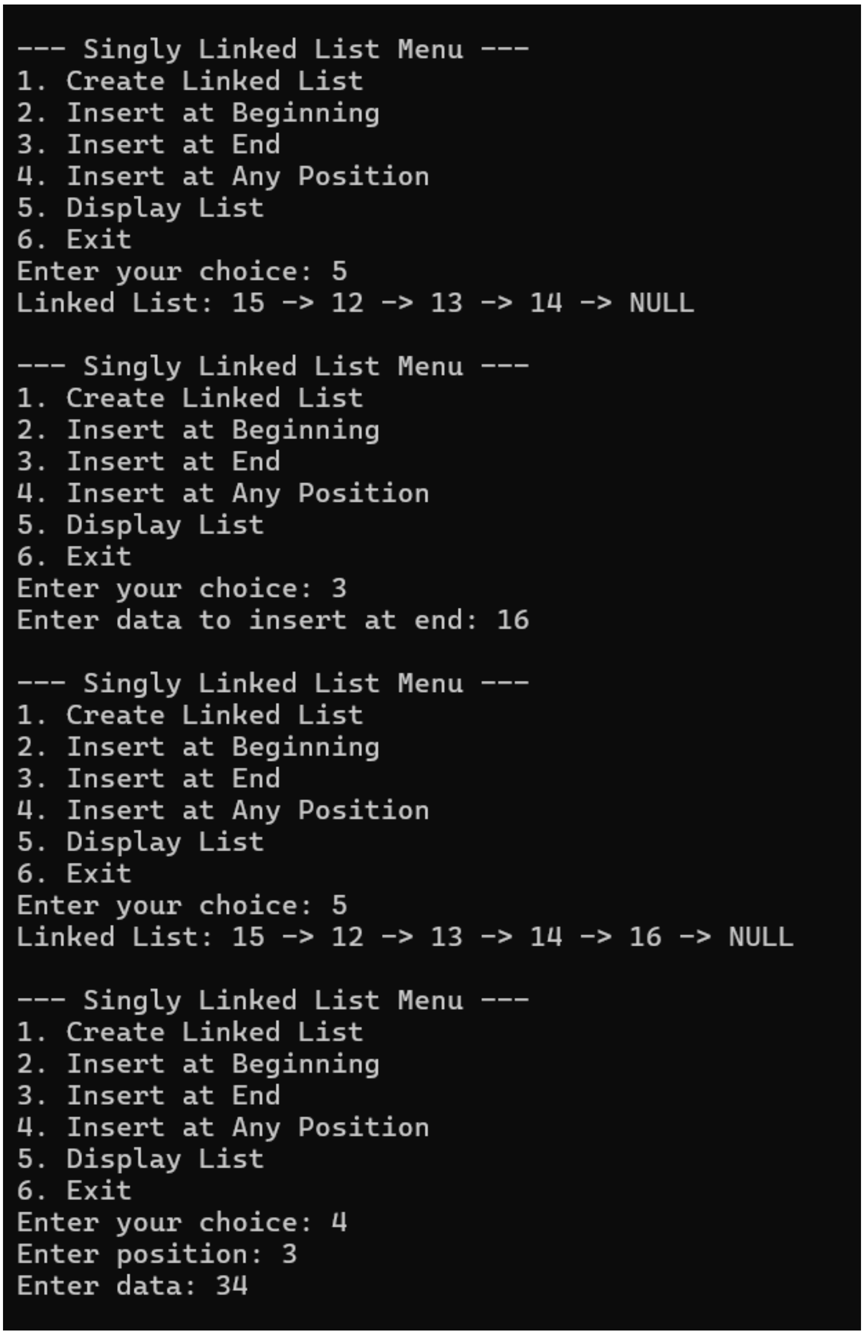
WAP to Implement Singly Linked List with following operations a) Create a linked list. b) Insertion of a node at first position, at any position and at end of list. Display the contents of the linked list.

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

// Definition of node
struct Node {
    int data;
    struct Node *next;
};

struct Node *head = NULL;

// Function to create a linked list
void createList() {
    int n, value;
    struct Node *temp, *newNode;

    printf("Enter number of nodes: ");
    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        newNode = (struct Node *)malloc(sizeof(struct Node));
        printf("Enter data for node %d: ", i + 1);
        scanf("%d", &value);

        newNode->data = value;
        newNode->next = NULL;

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

// Insert at beginning
void insertAtBeginning() {
    struct Node *newNode;
    int value;

    newNode = (struct Node *)malloc(sizeof(struct Node));
    printf("Enter data to insert at beginning: ");
    scanf("%d", &value);

    newNode->data = value;
    newNode->next = head;
    head = newNode;
}

// Insert at end
void insertAtEnd() {
    struct Node *newNode, *temp;
    int value;

    newNode = (struct Node *)malloc(sizeof(struct Node));
    printf("Enter data to insert at end: ");
    scanf("%d", &value);
    newNode->data = value;
    newNode->next = NULL;

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

// Insert at any position
void insertAtPosition() {
    struct Node *newNode, *temp;
    int value, pos, i = 1;

    printf("Enter position: ");
    scanf("%d", &pos);

    newNode = (struct Node *)malloc(sizeof(struct Node));
    printf("Enter data: ");
    scanf("%d", &value);

    newNode->data = value;

    if (pos == 1) {
        newNode->next = head;
        head = newNode;
        return;
    }
    temp = head;
    while (temp != NULL && i < pos - 1) {
        temp = temp->next;
        i++;
    }

    if (temp == NULL) {
        printf("Invalid position!\n");
    } else {
        newNode->next = temp->next;
        temp->next = newNode;
    }
}

// Display linked list
void display() {
    struct Node *temp;

    if (head == NULL) {
        printf("Linked List is empty.\n");
        return;
    }

    temp = head;
    printf("Linked List: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}
```

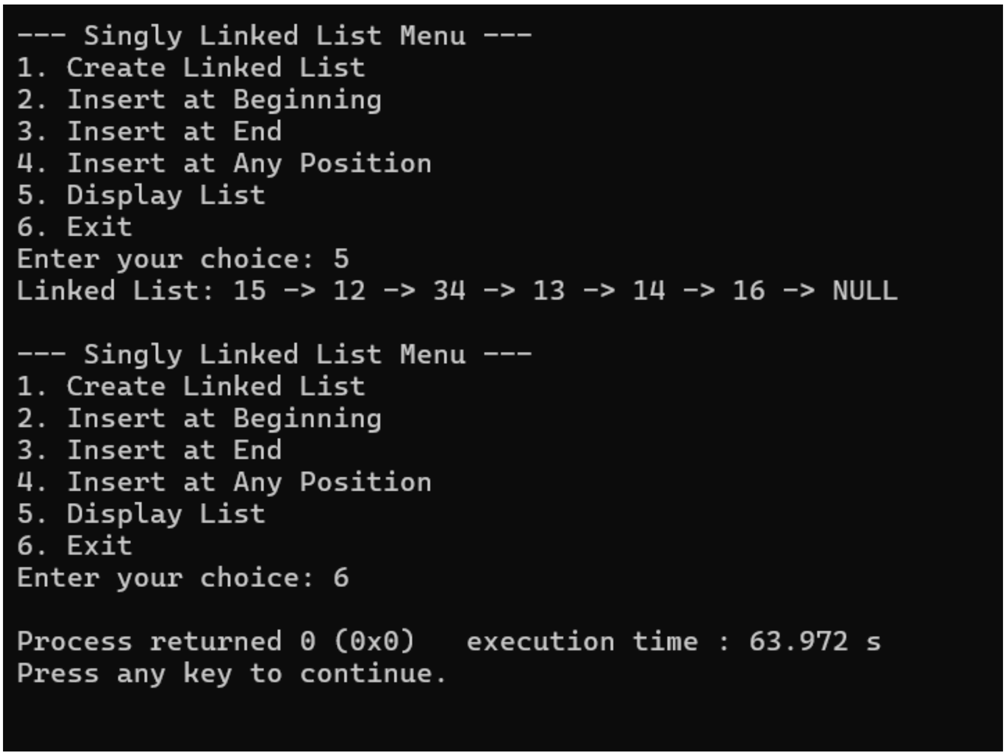
```
// Main menu
int main() {
    int choice;

    while (1) {
        printf("\n--- Singly Linked List Menu ---\n");
        printf("1. Create Linked List\n");
        printf("2. Insert at Beginning\n");
        printf("3. Insert at End\n");
        printf("4. Insert at Any Position\n");
        printf("5. Display List\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: createList(); break;
            case 2: insertAtBeginning(); break;
            case 3: insertAtEnd(); break;
            case 4: insertAtPosition(); break;
            case 5: display(); break;
            case 6: exit(0);
            default: printf("Invalid choice!\n");
        }
    }
    return 0;
}
```



```
--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at End
4. Insert at Any Position
5. Display List
6. Exit
Enter your choice: 1
Enter number of nodes: 3
Enter data for node 1: 12
Enter data for node 2: 13
Enter data for node 3: 14

--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at End
4. Insert at Any Position
5. Display List
6. Exit
Enter your choice: 5
Linked List: 12 -> 13 -> 14 -> NULL

--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at End
4. Insert at Any Position
5. Display List
6. Exit
Enter your choice: 2
Enter data to insert at beginning: 15
```



```
--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at End
4. Insert at Any Position
5. Display List
6. Exit
Enter your choice: 5
Linked List: 15 -> 12 -> 13 -> 14 -> NULL

--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at End
4. Insert at Any Position
5. Display List
6. Exit
Enter your choice: 3
Enter data to insert at end: 16

--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at End
4. Insert at Any Position
5. Display List
6. Exit
Enter your choice: 5
Linked List: 15 -> 12 -> 13 -> 14 -> 16 -> NULL

--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at End
4. Insert at Any Position
5. Display List
6. Exit
Enter your choice: 4
Enter position: 3
Enter data: 34
```



```
--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at End
4. Insert at Any Position
5. Display List
6. Exit
Enter your choice: 5
Linked List: 15 -> 12 -> 34 -> 13 -> 14 -> 16 -> NULL

--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at End
4. Insert at Any Position
5. Display List
6. Exit
Enter your choice: 6

Process returned 0 (0x0)   execution time : 63.972 s
Press any key to continue.
```