```
Enter number of initial nodes to create: 1
Enter 1 elements: 10

Linked List Menu
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Exit

Enter your choice: 1
Enter data: 5

Enter your choice: 2
Enter data: 15

Enter your choice: 3
Enter data: 20
Enter pos: 1

Enter your choice: 4
Linked List: 20 5 10 15

Enter your choice: 5

Process returned 0 (0x0)   execution time : 32.711 s
Press any key to continue.
```

LAB Program 4

Page No.
Date

Write a program to implement singly linked list and perform the following function.

a) Create a linked list

b) insertion of node at a) any position
b) fixed position
c) end of line

c) Display the contents of linked list

- ~~Algorithm~~ Linked List Psuedocode

```
STRUCT Node
    data
    next
end STRUCT

Function createNode(value)
    newnode = new Node
    newnode.data = value
    newnode.next = null
    return newnode
end function

function insert(head, value)
    new Node = createNode(value)
    newNode.next = head
    head = newNode
end function
```

```
function display (head)
    i = head
    while i != Null
      print i.data
      i = i.next
    end while
end function

main
   head = null
   insert (head, 10)
   display (head)
end main
```

Program

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

struct Node
 {
   int data;
   struct Node* next;
 };

struct Node* head = NULL;

void create (int n)
 {
   struct Node *newNode, *temp;
   int data, i;
   if (n<=0) return;
   for (i=0; i<n; i++)
   {
     newNode = (struct Node*) malloc (size of
     scanf (struct Node));
     scanf ("%d", &data);
     newNode -> data = data;
     newNode -> next = NULL;
     if (head == NULL)
        head = newNode;
     else
     {
       temp = head;
       while (temp -> next != NULL)
            temp = temp -> next;
```

```
        temp -> next = newNode;
      }
    }
}
void insertAtBeginning (int data)
{
    struct Node* newNode = (struct Node*) malloc
                          (sizeof(struct Node));
    newNode -> data = data;
    newNode -> next = head;
    head = newNode;
}

void insertAtEnd (int data)
  {
    struct Node *newNode, *temp;
    newNode = (struct Node*) malloc (sizeof(
                struct Node));
    newNode -> data = data;
    newNode -> next = NULL;
    if (head == NULL)
        head = newNode;
    else
      {
        temp = head;
        while (temp -> next != NULL)
            temp = temp -> next;
        temp -> next = newNode;
      }
}
```

```
void insertAtPosition (int data, int pos)
{
    int i;
    struct Node* newNode, *temp;
    newNode = (struct Node*) malloc (sizeof(
                struct Node));
    NewNode -> data = data;
    if (pos == 1)
    {
      newNode -> next = head;
      head = newNode;
      return;
    }
    temp = head;
    for (i=1; i < pos-1 && temp != NULL; i++)
        temp = temp -> next;
    if (temp == NULL) return;
    newNode -> next = temp -> next;
    temp -> next = newNode;
}

void display()
{
    struct Node* temp = head;
    while (temp != NULL)
    {
      printf ("%d", temp -> data);
      temp = temp -> next;
    }
    print("\n");
}
```

```
int main()
{
    int n, choice, data, pos;
    printf("Enter number of initial nodes to
            create:");
    scanf("%d", &n);
    printf("Enter %d elements:", n);
    create(n);
    printf("\n Linked List Menu\n 1. Insert
    at Beginning\n 2. Insert at End\n 3.
    Insert at Position\n 4. Display\n 5. Exit
    \n");

    while(1)
    {
     printf("\n Enter your choice:");
     scanf("%d", &choice);
     switch(choice)
     {
      case 1:
         printf("Enter data: ");
         scanf("%d", &data);
         insertAtBeginning(data);
         break;

      case 2:
         printf("Enter data:");
         scanf("%d", &data);
         insertAtEnd(data);
         break;

      case 3:
         printf("Enter data: ");
         scanf("%d", &data);
         printf("\n Enter pos");
         scanf("%d", &pos);
         insertAtPosition(data, pos);
         break;

      case 4:
         printf("Linked List:");
         display();
         break;

      case 5:
         exit(0);

      default:
         printf("Invalid choice\n");
     }
    }
    return 0;
    }
```

Output

```
Enter number of initial nodes to create: 1
Enter 1 elements: 10

Linked List Menu
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display
5. Exit
```

Enter your choice : 1
Enter data : 5

Enter your choice : 2
Enter data : 15

Enter your choice : 3
Enter data : 20
Enter pos : 1

Enter your choice : 4
Linked List : 20 5 10 15

Enter your choice : 5

10/11/25
o/p