operations:

a) Create a linked list

b) Insertion of a node at 1st position, at any position, end of the list.

c) Display the contents of linked list.

Pseudo code:

```
Struct Node {
    int data;
    Struct Node * next;
};

Void createlinkedlist (int n)
    head = Null;
    for (i = 1 ; i <= n; i++){
    newNode = (structNode *) malloc (size of (struct Node
    printf ( Enter data for node);
    scanf (%d , &data);
    new Node -> data = data;
    new Node -> Next = NULL;
```


```
if head == Null
head = newnode
else
temp -> next = new
temp = new node
```

```
void insert at Beginning (int value) {
    struct Node * newNode = (struct Node*)
                malloc (size of (struct Node));
    head = newNode;
    new node -> data = value;
    new node -> next = head;
}
```

```
void insert AtEnd(int value) {
    struct Node * newnode = (struct Node*) malloc
                            (sizeof(struct Node));
    Struct Node * temp;
    newnode -> data = value;
    newnode -> next = NULL;
    if (head == NULL)
        head = new Node
    else while (temp -> next != n
        temp = temp -> next;
        -> newnode
void insert AtPosition (int value, int pos) {
    if (pos < 1) : Return
    Create node * newnode
        if (pos == 1)
            { insert at beginning (value); }
    Struct Node * temp = head;
    for (int i = 1; i < pos - 1 && temp != Null;
                                    i++)
        temp = temp -> Next
    if (temp == Null) {
        free (Node);
    return; }
void display {
    struct Node * temp = head
        if (head == NULL)
            list is empty
        while (temp != NULL)
            temp -> data
            temp = temp -> next
        }
```

Code:

```
#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 (sixe of
                                  (Struct Node));
      scanf ("%d ", &data);
      new Node -> data = data;
      new Node -> next = NULL;
       if (head == NULL)
         head = newNode;
      else
       {
        temp = head;
        while (temp -> next != NULL)
           temp = temp -> next;
          temp -> next = new Node;
    }
 }
```

```
void insert At Beginning (int data)
{ struct Node *newNode = (struct Node *) malloc
                          (size of (struct Node));
  newNode -> data = data;
  newNode -> next = head;
  head = newNode;
}
void insert At End (int data)
{ Struct Node * newNode, * temp;
  newNode = (struct Node *) malloc (size of (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 insert At Position (int data, int pos)
{
  int i;
  struct Node * newNode, * temp;
  newNode = (struct Node*) malloc (size of
                                   (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;
}
printf("\n");
}
int main () {
int n, choice, data, pos;
printf ("Enter number of intial nodes
        to create : ");
scanf ("%d", &n);
printf ("Enter %d elements:", n);
create (n);
printf ("\n Linked list Menu\n1. Insert at
```

```
Beginning ... Insert at end \n3. Insert at
position \n4. Display \n5. Exit\n");
while (1) {
printf ("\n Enter your choice : ");
Scanf ("%d", &choice);
Switch (choice) {
case 1:
    printf ("Enter data:");
    Scanf ("%d", &data);
    insert At Beginning (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);
    insert At Position (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 : 4
Enter 4 elements : 1 2 3 4

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 : 8

Enter your choice : 2
Enter data : 7

Enter your choice : 3
Enter data : 6

Enter pos 3

Enter your choice : 4
Linked List : 8 1 6 2 3 4 7
Enter your choice : 5
Process returned 0
```