Lab 4 programs :-

Singly linked list

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

typedef struct node {
        int data;
        struct node* next;
    } node;
{
Node* head = NULL;

void push();
void append();
void insert();
void display();

void main()
{
    int ch;
    while(1)
    {
        printf("1. Insert at beginning \n");
        printf("2. Insert at end \n");
        printf("3. Insert at position \n");
        printf("4. Display \n");
        printf("5. Exit \n");
        printf("Enter choice :");
        scanf("%d", &ch);
        switch(ch)
        {
            case 1: push();
                    break;
```

Date __/__/__
Page _____

```
Case 2:
    append();
    break;
case 3:
    insert();
    break;
case 4: display();
        break;
default: printf("Existing");
}
}
}
void push()
{
    Node *temp = (Node*) malloc(sizeof(Node));
    int new_data;
    printf("Enter data in new node:");
    scanf("%d", &new_data);
    temp -> data = new_data;
    temp -> next = head;
    head = temp;
}
void append()
{
    Node *temp = (Node*) malloc(sizeof(Node));
    int new_data;
    printf("Enter data:");
    temp -> data = new_data;
    temp -> next = NULL;
    if (head == NULL)
    {
        head = temp;
    return;
```

```
Node* temp1 = head;
while (temp -> next != NULL)
    {
        temp1 = temp1 -> next;
    }.
temp1 -> next = temp;
}.
void insert ()
{
Node * temp = (Node*) * malloc (sizeof (Node));
int new_data, pos;
printf ("Enter data");
scanf ("%d", &new_data);
printf ("Enter position:");
scanf ("%d", &pos);
temp -> data = new_data;
temp -> next = NULL;
if (pos == 0).
    temp -> next = head;
    head = temp;
    return;
}.
Node * temp1 = head;
while (pos --)
    {
        temp1 = temp1 -> next;
    }
Node * temp2 = temp1 -> next;
temp -> next = temp2;
temp -> next = temp;
}.
```

Date __/__/__
Page _____

```
void display ()
{
    Node * temp1 = head;
    while (temp != = NULL)
    {
        printf (" %d ->", temp1 -> data);
    }
    printf ("NULL");
}
```

OUTPUT:

Enter choice : 1

Enter data in new node : 0

1. Insert at beg
2. Insert at end
3. Insert at pos
4. Display
5. Exit

Enter choice : 2

Enter data : 1

Enter position of new node : 1

1. Insert at beg
2. Insert at end
3. Insert at pos
4. Display
5. Exit

Enter choice : 4

0 -> 2 -> 1 -> NULL

5/11/24