

LAB PROGRAM - 8 i) (Linked list as a Stack).
ii) (Linked list as a queue).

```
(1) #include <stdio.h>
#include <stdlib.h>
void push();
void pop();
void display();
struct node
{
    int data;
    struct node *next;
};
struct node *top = NULL;

int main()
{
    int choice;
    do
    {
        printf("\n --- Stack Menu --- \n");
        printf("\n 1. Push 2. Pop 3. Display 4. EXIT");
        printf("\n Enter your choice : \n");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1: push(); break;
            case 2: pop(); break;
            case 3: display; break;
            case 4: printf("EXITING... \n"); break;
            default: printf("INVALID choice \n");
        }
    } while (choice != 4);
    return 0;
}
```

```
void push()
```

```
{
```

```
    int item;
```

```
    struct node *newnode;
```

```
    printf("Enter element you want to push \n");
```

```
    scanf("%d", &item);
```

```
    newnode = (struct node *) malloc (sizeof (struct node));
```

```
    newnode->data = item;
```

```
    newnode->next = NULL;
```

```
    if (top == NULL)
```

```
    { top = newnode;
```

```
    } else {
```

```
        newnode->next = top;
```

```
        top = newnode;
```

```
    }
```

```
}
```

```
void pop()
```

```
{
```

```
    if (top == NULL)
```

```
    { printf("Stack is empty \n");
```

```
    } else {
```

```
        printf("Element removed is : %d \n", top->data);
```

```
        top = top->next;
```

```
    }
```

```
}
```

```
void display()
```

```
{
```

```
    struct node *temp;
```

```
    temp = top;
```

```
    if (top == NULL)
```

```
    {
```

```
        printf("Stack is empty \n");
```

```
        return;
```

```
    }
```



```

printf("Contents of Stack are : \n");
while (temp != NULL)
{
    printf("%d\t", temp->data);
    temp = temp->next;
}
printf("\n");
}

```

```

(ii) #include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *next;
};

```

```

void insert();
void display();
void delete();
struct node *rear = NULL, *front = NULL;
int main()
{

```

```

    int choice;
    do {
        printf("\n - - - MENU - - - \n");
        printf("\n1. Create\n2. Delete\n3. Display\n4. EXIT ");
        printf("\nEnter your choice : \n");
        scanf("%d", &choice);
        switch(choice)
        {

```

```

            case 1: insert(); break;
            case 2: delete(); break;

```

```

        case 3: display(); break;
        case 4: printf("EXITING....\n"); break;
        default: printf("Invalid choice\n");
    }
    while (choice != 4);
    return 0;
}

```

```

void insert()
{

```

```

    struct node *newnode;
    newnode = (struct node *) malloc(sizeof(struct node));

```

```

    printf("Enter the element: \n");

```

```

    scanf("%d", &newnode->data);

```

```

    newnode->next = NULL;

```

```

    if (rear == NULL)
    {

```

```

        rear = newnode;

```

```

        front = newnode;
    }

```

```

    else {

```

```

        rear->next = newnode;

```

```

        rear = newnode;
    }
}

```

```

void delete()
{

```

```

    if (front == NULL)
    {

```

```

        printf("Queue Underflow\n");
        return;
    }

```

```

    else {

```



```
printf("Deleted Element is: %d\n",  
      front->data);
```

```
if (front == rear)
```

```
{  
    front = NULL;  
    rear = NULL;
```

```
}
```

```
else
```

```
    front = front->next;
```

```
}
```

```
}
```

```
void display()
```

```
{
```

```
    struct node *temp;
```

```
    if (front == NULL)
```

```
{
```

```
        printf("Queue is Empty!\n");
```

```
        return;
```

```
}
```

```
    temp = front;
```

```
    printf("Contents of queue are: \n");
```

```
    while (temp != NULL)
```

```
{
```

```
        printf("%d\t", temp->data);
```

```
        temp = temp->next;
```

```
}
```

```
    printf("\n");
```

```
}
```