

Lab 8

Dhanyashree K
IBM19C3054

```
#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;
    printf("\n Stack implementation using linked list\n");
    do
    {
        printf("\n 1. Push 2. Display 3. Pop\n 4. Exit\n");
        printf("\n Enter your choice;");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1: push(); break;
            case 2: display(); break;
            case 3: pop(); break;
            case 4: exit(0);
        }
    } while (choice != 4);
}

void push()
{
    int item;
    struct node * newnode;
```

```

printf("Enter the element\n");
scanf("%d", &item);
newnode = (struct node*)malloc(sizeof
(struct node));

```

```

newnode->data = item;
newnode->next = NULL;
if (top == NULL)
    top = newnode;
}

```

```

void pop()
{
    if (top == NULL)
        printf("stack is empty");
    else
    {
        printf("element removed is %d", top->data);
        top = top->next;
    }
}

```

```

void display()
{
    struct node *temp;
    temp = top;
    if (top == NULL)
        printf("Stack is empty");
    while (temp != NULL)
    {
        printf("%d", temp->data);
        temp = temp->next;
    }
}

```

```

#include <stdio.h>
#include <stdlib.h>
struct node
{

```

```

int data;
struct node * next;
};

void insert ();
void display ();
void del ();
struct node * rear = NULL, * front = NULL;
int main (int argc, char ** argv)
{
    printf ("Linked List Implementation using linked list");
    int choice;
    do
    {
        printf ("1. Create 2. Display 3. Delete 4. Exit\n");
        printf ("Enter your choice: ");
        scanf ("%d", &choice);
        switch (choice)
        {
            case 1: insert (); break;
            case 2: display (); break;
            case 3: del (); break;
            case 4: exit (0);
        }
    } while (choice != 4);
}

void insert ()
{
    struct node * newnode;
    newnode = (struct node *) malloc (sizeof (struct node));
    printf ("Enter the element: ");
    scanf ("%d", &newnode->data);
    newnode->next = NULL;
    if (rear == NULL)

```



```

    rear = newnode;
    front = newnode;
}
else
{
    rear->next = newnode;
    rear = newnode;
}
}

```

```

void del ()
{

```

```

    if (front == NULL)
    {
        printf("Queue is empty\n"); return;
    }

```

```

    else
    {

```

```

        printf("Delete Deleted element is %d", front
            → data);

```

```

        if (front == rear)

```

```

        {
            printf("Deleted element is

```

```

                printf("\n Queue is empty\n");

```

```

                front = NULL; rear = NULL;
            }

```

```

        else

```

```

            front = front->next

```

```

        }
    }

```

```

void display
{

```

```


```

```

    struct node *temp;

```

```

    if (front == NULL)
    {

```

```


```

```

        printf("Queue is empty\n");

```

```

        return;
    }

```

```

}

```

```
temp = front;  
while (temp != NULL)  
{  
    printf ("%d", temp->data);  
    temp = temp->next;  
}
```

Durgam.