Lab-8

```
struct node
{
    int data;
    struct node *next;
};
void insert()
struct node *rear = NULL, *front = NULL;
struct node *top = NULL;

void push()
{
    int item;
    struct node *newnode;
    printf("Enter the element")
    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");
    else
    {
        printf("element removed is %d", top->data);
        top = top->next;
    }
}
void display_stack()
{
    struct node *temp;
    temp = top;
    if (top == NULL)
        printf("Stack is empty");
    while(temp != NULL)
    {
        printf("%d", temp->data);
        temp = temp->next;
    }
}
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 de is %d", front->data);
      if(front == rear)
      {
        printf("Queue is empty");
        front = NULL;
        rear = NULL;
      }
      else
      {
        front = front->next;
      }
}
```

```
void display_q()
{
    struct node *temp;
    if(front == NULL)
    {
        printf("Queue is empty");
        return;
    }
    temp = front;
    while(temp != NULL)
    {
        printf("%d", temp->data);
        temp = temp->next;
    }
}
```