DHRUVA M
1BM19CS04

```
void push(struct node **top, int d)
{
 struct node *temp, n;
 temp = (struct node *) malloc(size of (struct node));
 if (temp == NULL)
 {
  printf("Stack is full\n");
 }
 temp -> data = d;
 temp -> next = *top;
  *top = temp;
  printf("%d is pushed\n");
}
void pop (struct node **top)
  {
    struct node *temp;
      if (*top == NULL)
    {
      printf("Stack Underflow");
      return;
    }
```

Scanned with CamScanner

```
        temp = *top;
        printf("%d poped\n", temp->data);
        *top = (*top)->next;
        free(temp);
    }
void display(struct node *top)
    {
        if(top == NULL)
        {
        printf("No elements in stack");
        return;
        }
        while(top != NULL)
        {
        printf("%d", top->data);
        top = top->next;
        }
        printf("\n");
    }
```

DHRUVA
1BM19CS0

Scanned with CamScanner

```
void insert (int d)
{
  struct node *n;
  n = (struct node *) malloc (size of (struct node));
  if (n == NULL)
  {
    printf ("Queue Overflow \n");
    return;
  }
  n -> data = d;
  if (front == NULL)
  {
    front = n;
    rear = n;
    front -> next = NULL;
    rear -> next = NULL;
  }
  else {
    rear -> next = n;
    rear = n;
    rear -> next = NULL;
  }
  printf ("%d is inserted");
}
```

DHRUVA M
1BM19CS04

```
void delete()
{
    struct node *temp;
    if (front == NULL)
    {
        printf("Queue Underflow");
        return;
    }
    temp = front;
    printf("%d deleted\n", temp->data);
    front = front->next;
    free(temp);
}
void display_queue()
{
    struct node *temp;
    temp = front;
    if (front == NULL)
    {
        printf("Empty queue\n");
    }
    else
    {
        printf("Queue elements :\n");
```

Scanned with CamScanner

DHRUVA M
1BM19CS049

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

Scanned with CamScanner