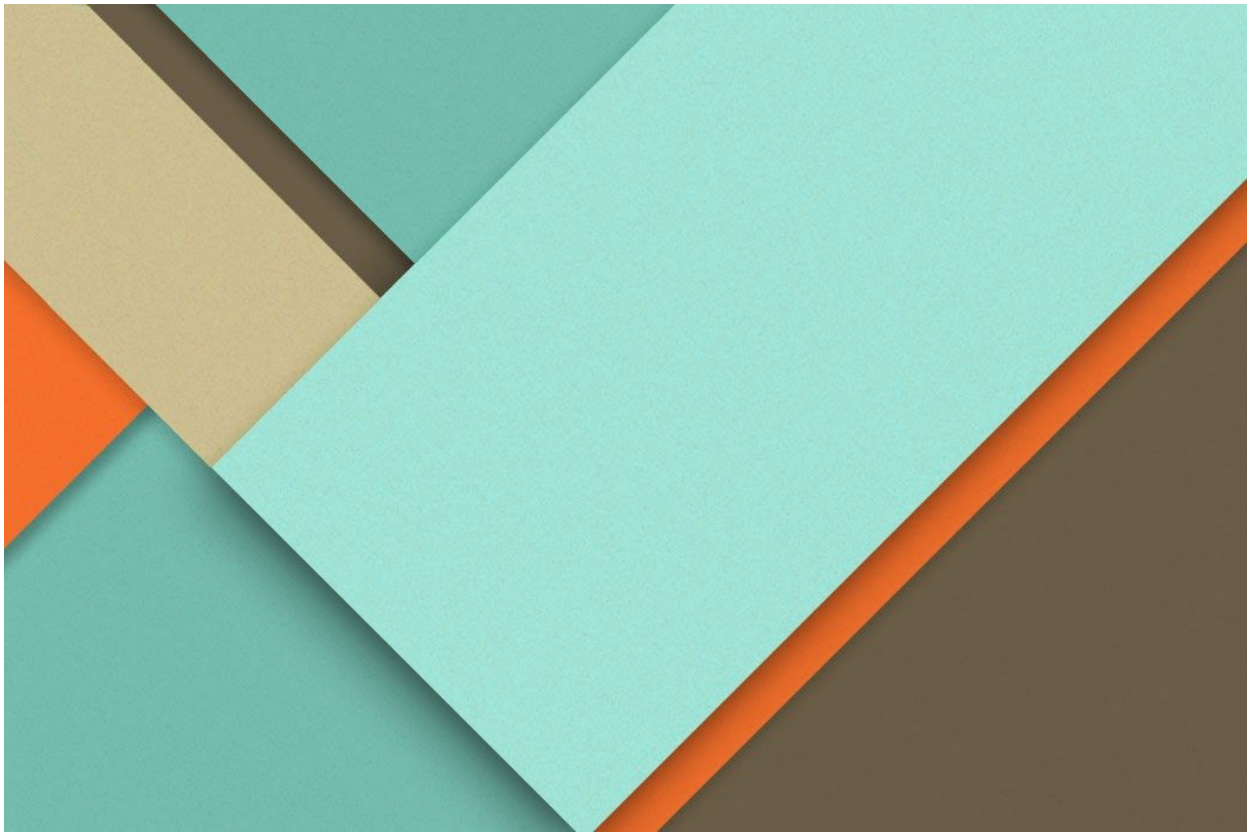

DSA Assignment

Boyapati Sai Venkat

AP19110010174

1st-year CSE-E.

Programs on Circular Queues

Write a menu-driven program to implement the following operations in a circular queue.

a. Create an empty circular queue

b. Insert an element into the circular queue

c. Delete an element from the circular queue. Display the content of the circular queue with a clear indication of Front and Rear element

Solution:

```
# include<stdio.h>
# define size 5

int queue[size],MAX;
int front =-1;
int rear =-1;
```

```
void insert(int item)
{
	if((front == 0 && rear == MAX-1) || (front == rear+1))
	{
		printf("Queue Overflow \n");
		return;
	}
	if (front == -1)
	{
		front = 0;
		rear = 0;
	}
	else
	{
		if(rear == MAX-1)
			rear = 0;
		else
			rear = (rear+1)/size ;
	}
	queue[rear] = item ;
}


void del()
{
	if (front ==-1)
	{
```

```
                printf("Queue Underflow\n");
                return ;
        }
        printf("Element deleted from queue is : %d\n",queue[front]);
        if(front == rear)
        {
                front = -1;
                rear=-1;
        }
        else
        {
                if(front == MAX-1)
                        front = 0;
                else
                        front = (front+1)/size;
        }
}


void display()
{
        int front_pos = front,rear_pos = rear;
        if(front ==-1)
        {
                printf("Queue is empty\n");
                return;
        }
        printf("Queue elements :\n");
        if( front_pos <= rear_pos )
```

```
                while(front_pos <= rear_pos)
                {
                        printf("%d ",queue[front_pos]);
                        front_pos++;
                }
        else
        {
                while(front_pos <= MAX-1)
                {
                        printf("%d ",queue[front_pos]);
                        front_pos++;
                }
                front_pos = 0;
                while(front_pos <= rear_pos)
                {
                        printf("%d ",queue[front_pos]);
                        front_pos++;
                }
        }
        printf("\n");
}


int main()
{
        int choice,item;
        do
        {
                printf("1.Insert\n");
```

```
        printf("2.Delete\n");
        printf("3.Display\n");
        printf("4.Quit\n");

        printf("Enter your choice : ");
        scanf("%d",&choice);

        switch(choice)
        {
            case 1 :
                printf("insert the element  in queue : ");
                scanf("%d", &item);

                insert(item);
                break;
            case 2 :
                del();
                break;
            case 3:
                display();
                break;
            case 4:
                break;
                default:
                printf("invalid choice!\n");
        }
}while(choice!=4);

return 0;
```

```
}
```

Output:

```
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice: 1
insert the element  in the queue : 1 0
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice: 1
insert the element  in the queue : 1
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice: 1
insert the element  in the queue : 2
```

1.Insert

2.Delete

3.Display

4.Quit

Enter your choice: 1

insert the element in the queue : 3

1.Insert

2.Delete

3.Display

4.Quit

Enter your choice: 1

insert the element in the queue : 4

1.Insert

2.Delete

3.Display

4.Quit

Enter your choice: 2

Element deleted from the queue is: 4

1.Insert

2.Delete

3.Display

4.Quit

Enter your choice : 3

Queue is empty

1.Insert

2.Delete

3.Display

4.Quit

Enter your choice: 4