Circular Queue

Code Id	18
Date Updated	3/7/2010
Title	Circular queue

Description

```
This is a program of circular queue representation using array.
```

Codes Snippet

```
# include
# define MAX 5

int cqueue_arr[MAX];
int front = -1;
int rear = -1;
main()
{
        int choice;
        while(1)
        {
                printf("1.Insertn");
                printf("2.Deleten");
                printf("3.Displayn");
                printf("4.Quitn");
                printf("Enter your choice : ");
                scanf("%d",&choice);

                switch(choice)
                {
                case 1 :
                        insert();
                        break;
                case 2 :
                        del();
                        break;
                case 3:
                        display();
                        break;
                case 4:
                        exit(1);
                default:
                        printf("Wrong choicen");
                }/*End of switch*/
        }/*End of while */
}/*End of main()*/
insert()
{
        int added_item;
        if((front == 0 && rear == MAX-1) || (front == rear+1))
        {
                printf("Queue Overflow n");
                return;
        }
        if (front == -1)  /*If queue is empty */
        {
                front = 0;
                rear = 0;
        }
        else
                if(rear == MAX-1)/*rear is at last position of queue */
                        rear = 0;
                else
                        rear = rear+1;
        printf("Input the element for insertion in queue : ");
        scanf("%d", &added_item);
        cqueue_arr[rear] = added_item ;
}/*End of insert()*/
del()
{
        if (front == -1)
```

```
                        front = 0;
                else
                        front = front+1;
}/*End of del() */
display()
{
        int front_pos = front,rear_pos = rear;
        if(front == -1)
        {
                printf("Queue is emptyn");
                return;
        }
        printf("Queue elements :n");
        if( front_pos <= rear_pos )
                while(front_pos <= rear_pos)
                {
                        printf("%d ",cqueue_arr[front_pos]);
                        front_pos++;
                }
        else
        {
                while(front_pos <= MAX-1)
                {
                        printf("%d ",cqueue_arr[front_pos]);
                        front_pos++;
                }
                front_pos = 0;
                while(front_pos <= rear_pos)
                {
                        printf("%d ",cqueue_arr[front_pos]);
                        front_pos++;
                }
        }/*End of else */
        printf("n");
}/*End of display() */
```