CIRCULAR QUEUE

DEREK STANLEY
KANNA THE
18MIS1005

```
int front = -1;
 int rear = -1;
 # define size 5
 void insert (int ele , int queue [])
  {
      if ((rear+1) % size == front)
         printf(" Queue Overflow");
      else
      {   rear = (rear+1) % size;
          if (front == -1)
             front = 0;
          queue[rear] = ele;
      }
  }

 int  delete ( int queue [])
 {   if ( rear == -1)
         return -9999;
     else
     {  item = queue (rear];
        if ( front==rear)
        { front = -1;
          rear = -1;
        } else { front =(front+1) % size ; }
       return item;
     }
 }
 void display (int queue[])
 {   if((front == -1) && (rear == -1))
       { printf(" Queue is empty"); }
     else
     {  ~~for ( int~~
        if (front <= rear )
        { for (int i =front ; i<=rear ; i++)
            printf(~~queue[i]~~ "%d \n", queue(i]);
        else
        { for(int i = front ; i<= size-1 ; i++)
            printf("%d\n", queue[i]);
          for ( int i=0 ; i<= rear ; i++)
            printf("%d\n", queue[i]);
        }
      }
 }
```