WAP to implement Priority Queue

```c
# define MAX5
int pri_Q [MAX];
int front = -1, rear = -1;

check (x)
{
for (int i=0; i <= rear; i++)
{
    if (x > = pri_Q [i])
    {
        for ( j = rear +1; j > i; j--)
        {
            pri_Q [j] = pri_Q [j-1];
        }
        pri_Q [i] = x;
        return;
    }
}
    pri_q [i] = x;
}

insert (x)
{
    if ( rear > = MAX-1)
    printf (" Queue Overflow")
    if ( front == -1 && rear == -1)
    {
        front ++;
        rear ++;
        pri_Q [rear] = x;
    }

    else
        check (x);
}        rear ++;
```

```c
delete (x)
{
    int i
    if ( front == -1 && rear == -1)
        printf (" Empty Queue")
    else
        for (i=0; i <= rear; i++)
        {
            if (x == pri_Q[i])
            {
                for ( ; i < rear ; i++)
                    pri_Q[i] = pri_Q[i+1];

                pri_Q[i] = -99;
                rear--;
                if (rear == -1)
                    front = -1
                return ;
            }
        } printf (" Element not found")
}

display ()
{
    if (front = rear = -1)
        printf (' Empty Queue")

    while (front <= rear)
    {
        printf (" %d", pri_Q[front]);
        front++;
    }
    front = 0;
}
```