Trisha Lakhani
1BM19CS214

Priority Queue

```
int  A[SIZE] ,   front , rear
create ()
{ front ← rear = -1
   }
insert-by-priority ( int data)
{
     if ( rear >= SIZE-1)
   { printf (" Queue Overflow");
      return ;
    }
     if ((front==-1)&&(rear= =-1))
      {
         front = front + 1;
         rear  =  rear + 1;
         A [rear] = data ;
       }
     else
          check (data);
          rear ++ ;
     }

void check (int data)
{
   int  i, j ;
   for  ( i=0 ;  i<= rear;  i++)
   {       if   (data >= A [i] )
          { for ( j= rear + 1 ; j > i , j--)
              {      A[j] = A[j-1];
               }
```

```
            A[i] = data;
            return;
          }
       }
            A[i] = data;
   }

void delete_by_priority (int data)
{    int i;
     if ((front == -1) && (rear == -1))
      {     printf ("Queue is empty");
            return;
      }
     for (i=0; i<= rear; i++)
      {
            if (data == A[i])
              {     for (; i< rear; i++)
                     {     A[i]= A[i+1];
                     }
                    A[i]= -99;
                    rear--;
                    if (rear == -1)
                        front = -1;
                    return;
              }
      }
     printf ("Element not found");
}
```

```
void display-queue ( )
{
    if ((front ==-1) && (rear== -1))
      {
         printf(" Queue is Empty');
         return;
      }
    for ( ; front <= rear; front++)
      {
         printf ("%d", A[front]);
      }
      front = 0;
}.
```