q 1   Quick Sort   21k 3278   Sohaib

```
int  Partition ( int arr[], int s , int e )
{

     int pivot = arr[s]

     int pivot_pos = s
  if ( s < e )
  {  while ( arr[s] <= pivot ) [illegible]
     {
          s++;
     }

     while ( arr[e] > pivot )
     {
          e [illegible];
     }

     swap ( arr[s] , arr[e] );
  }

    swap ( arr[pivot_pos] , arr[e] ) ,
    return e ;
}


  void  quickSort ( int arr[], int s , int e )
  {
      if ( s > e )
      { }  ⟶  return;

      int P = Partition ( arr , s , e )
      quickSort ( arr , s , p-1 );
      quickSort ( arr , p+1 , e );
  }
```

q 1   Radix Sort   21k-3278

```
void   CountSort ( int arr[], int s ,  int pos)
{    int  count [10] = { 0 }
     int  output [10] = { 0 }

     for (int i=0 , i<s , i++)
            count [(arr[i] / pos) % 10]++;
     for ( int i=1 ; i<10 ; i++)
            count[i]  = count [i] + count [i-1] ;
     for ( int i= size-1 , i>=0 ; i--)
            output [--count[(arr[i] / pos) % 10]] = arr [i] ;

     for ( int i=0 ; i<s ; i++)
            arr [i] = output [i] ;
}


void  Radix Sort ( int  arr[] , int  s )
{

   int  max = Find_Max ( arr, size )

   for (int pos=1 ,  max/pos > 0 ;  pos *= 10)

          CountSort ( arr , size , poc) ;

}


int  Find_max ( int *arr , int s )
{   max = arr [0] ;
    for (int i=0 , i< s ; i++)
      {  if ( arr [i] > max)
             max = arr [i]
      }
}
```

q2 2-1K 3278

```
#define N 5
# inclde <iostream>

void enqueue (int queue[], int x , int& front , int &rear)
{
        if (rear == N-1)
           {    cout<<"overflow" ;}
        else if ( front == -1 && rear == -1)
             {  front = rear = 0;        queue [rear] = x ; }
        else }
                rear++;
                queue[rear] = x;
                }
}


void dequeue (int queue[], int &front , int &rear).
{
      if ( front == -1 && rear == -1)
          {   cout<< "underflow"; }
      else if (front == rear).
           {  front = rear -1 ; }
      else {
               cout<< queue [front];
               front ++;
         }
}


void peek (int queue[], int & front , int &rear).
{
        if ( front == -1 && rear == -1).
          { coutcc "underflow" }.
        else{
               coutcc. queue [front];
          }
}

void print ( ).
{
      for (int i = front ; i< rear+1 ; i++).
        { coutcc queue[i] ; }
}
```