

Part 2: Pseudocode and explanation

//We run a single Input for, Number of runways (part 1, step 1, 1st input)

```
cin>>runway;
```

// Next, we run a for Loop for Input, delay time for runway from 0 to runway (part 1, step 2, 2nd input)

```
for(int i=0;i<runway;i++) {
```

```
    cin>>rdelay[i];}
```

//Next, we run a for loop for Input, capacity of each runway mean Airplane can stand in a runway from 0 to runway (part 1, step 3, 3rd Input)

```
for (int i=0; i<runway;i++){
```

```
    cin>>Capacity[i];}
```

//Next, we run a single Input for, Total number of planes (part 1, step 4, 4th Input)

```
cin>>plane;
```

//Next, we run a for loop for Input, delay time for each plane 0 to plane (part 1, step 5, 5th Input)

```
for (int i=0; i<plane;i++){
```

```
    cin>>pdelay[i];}
```

//We use enqueue () function to insert the delay times into the queue, so that when the 1st plane will be take off from the runway it's delay time will be used for calculate Minimum total delay (part 2, step 1)

```
Airport pDelay;
```

```
for (int i=0;i<=plane;i++){
```

```
    pDelay.enqueue(pdelay[i]);}
```

//using dequeue() functionwe remove the planes from runway track(part 2, step 2)

//Function: getSize(), getSum(),getMax()(part 3, step1)

//using these functions, we find out the maxTotaldelay forparticular Inputes...

```
TotalDelay(Airport Q[], int n, int pDelay, int rIndex)
```

```
{
```

```

int max=0,sum;
for (int i=0;i<n;i++){
    if(rIndex==i){
        sum=Q[i].getSum()+pDelay;
    } else{ sum=Q[i].getSum();}
    if(max<sum){max=sum;}
}
return max;}

//Planes are taking off from which runways and their delays (part 4)
Airport Q[runway];
while (pDelay.getSize()>0)
{
for (int i=0; i<runway; i++){
    for (int j=0;j<runway;j++){
        if (Capacity[j]>Q[j].getSize() && rdelay[j].delay>=0){
            int mtd = TotalDelay(Q,runway,delay,j);
if(mTDelay>mtd;{
                mTDelay=mtd;
                mTDelayIn=j;}}}}
        if(rdelay[i].delay>=0 && mDelay>rdelay[i].delay && Capacity[i]>Q[i].getSize()){
            mDelay = rdelay[i].delay;
            mDelayIn= i;} }
if(TotalDelay(Q,runway,delay,mDelayIn)<TotalDelay(Q,runway,delay,mTDelayIn);{
    cout<<"Plane delay #"<<delay<<" at Runway delay #"<<rdelay[mDelayIn]<<endl;
    Q[mDelayIn].enqueue(delay);}else{
    cout<<"Plane delay #"<<delay<<" at Runway delay #"<<rdelay[mTDelayIn]<<endl;
    Q[mTDelayIn].enqueue(delay); }
cout<<endl;}

```

****After the iteration is completed, we should have our desired results.**