Name: Agoorukasetty Adithya

Email: adithyaak18@gmail.com

Question:

Airplane Seating Algorithm

Write a program that helps seat audiences in a flight based on the following input and

rules.

Rules for seating

1. Always seat passengers starting from the front row to back, starting from the left to

the right

2. Fill aisle seats first followed by window seats followed by center seats (any order in

center seats)

Input to the program will be

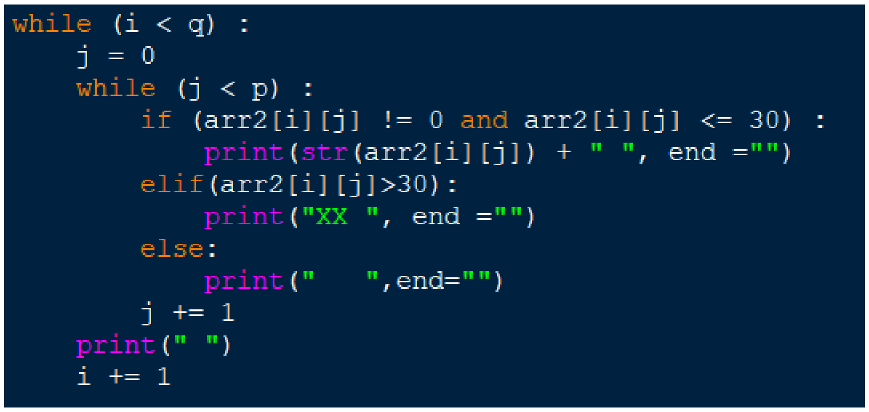
1. A 2D array that represents the columns and rows - Ex. [[3,4], [4,5], [2,3], [3,4]]

2. Number of passengers waiting in the queue.

Python code:

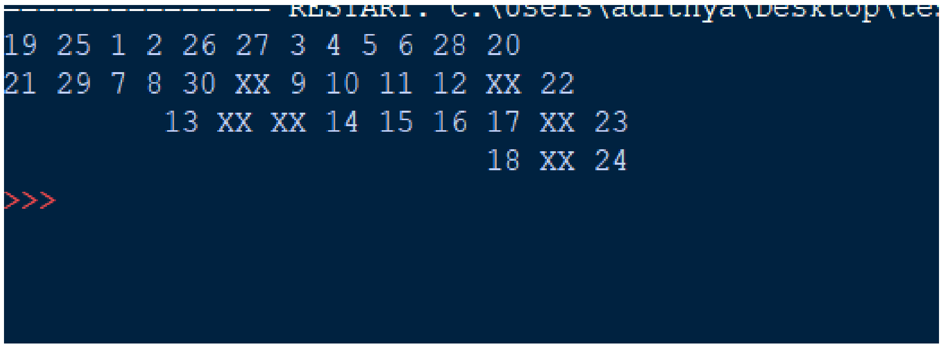
```python
arr = [[3, 2], [4, 3], [2, 3], [3, 4]]
n = len(arr)
count = 1
k = 30
p = 0
q = -1
aisle =  []
aisle.append(arr[0][0] - 1)
sum = arr[0][0] - 1
i = 1
while (i < n - 1) :
    if (arr[i][0] != 2) :
        sum += 1
        aisle.append(sum)
        sum += arr[i][0] - 1
        aisle.append(sum)
    else :
        sum += 1
        aisle.append(sum)
        sum += 1
        aisle.append(sum)
    i += 1
aisle.append(sum + 1)
i = 0
while (i < n) :
    p += arr[i][0]
    if (arr[i][1] > q) :
        q = arr[i][1]
    i += 1
window =  []
window.append(0)
window.append(p - 1)
center =  []
i = 0
while (i < p) :
    if (not i  in aisle and not i  in window) :
        center.append(i)
    i += 1
arr2 = [[0] * (p) for _ in range(q)]
i = 0
while (i < q) :
    j = 0
    while (j < p) :
        arr2[i][j] = 0
        j += 1
    i += 1
sum2 = -1
```

```python
limit =  []
limit.append(0)
i = 0
while (i < n) :
    sum2 += arr[i][0]
    limit.append(sum2)
    i += 1
map =  dict()
map[0] = arr[0][1]
i = 0
while (i < n) :
    j = limit[i] + 1
    while (j <= limit[i + 1]) :
        map[j] = arr[i][1]
        j += 1
    i += 1

i = 0
while (i < q) :
    j = 0
    while (j < p) :
        if (j  in aisle and i < map.get(j)) :
            arr2[i][j] = count
            count += 1
        j += 1
    i += 1
i = 0
while (i < q) :
    j = 0
    while (j < p) :
        if (j  in window and i < map.get(j)) :
            arr2[i][j] = count
            count += 1
        j += 1
    i += 1
i = 0
while (i < q) :
    j = 0
    while (j < p) :
        if (j  in center and i < map.get(j)) :
            arr2[i][j] = count
            count += 1
        j += 1
    i += 1
i = 0
while (i < q) :
    j = 0
```



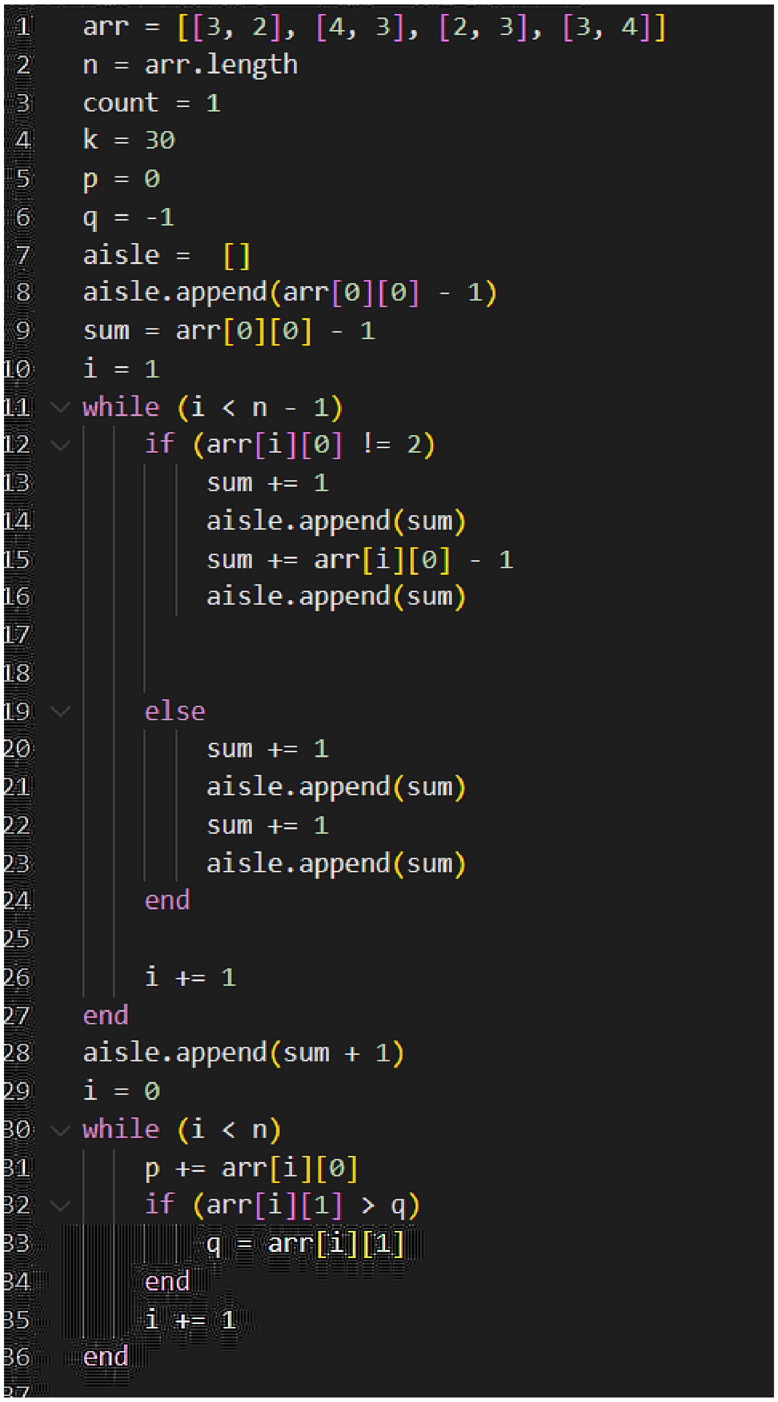
```python
while (i < q) :
    j = 0
    while (j < p) :
        if (arr2[i][j] != 0 and arr2[i][j] <= 30) :
            print(str(arr2[i][j]) + " ", end ="")
        elif(arr2[i][j]>30):
            print("XX ", end ="")
        else:
            print("   ",end="")
        j += 1
    print(" ")
    i += 1
```

Output:

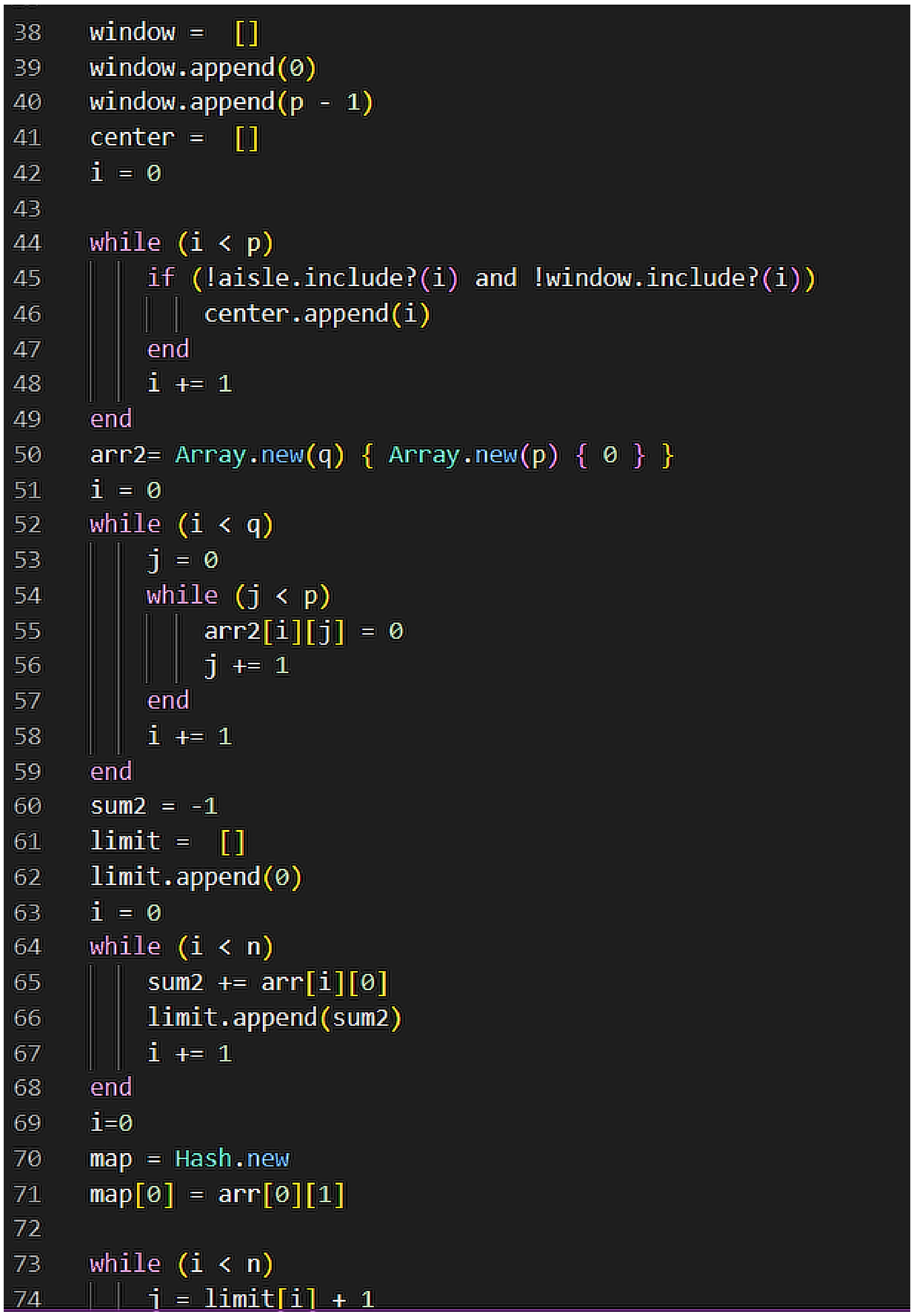


```
================ RESTART: C:\Users\adithya\Desktop\te
19 25 1 2 26 27 3 4 5 6 28 20
21 29 7 8 30 XX 9 10 11 12 XX 22
         13 XX XX 14 15 16 17 XX 23
                           18 XX 24
>>>
```

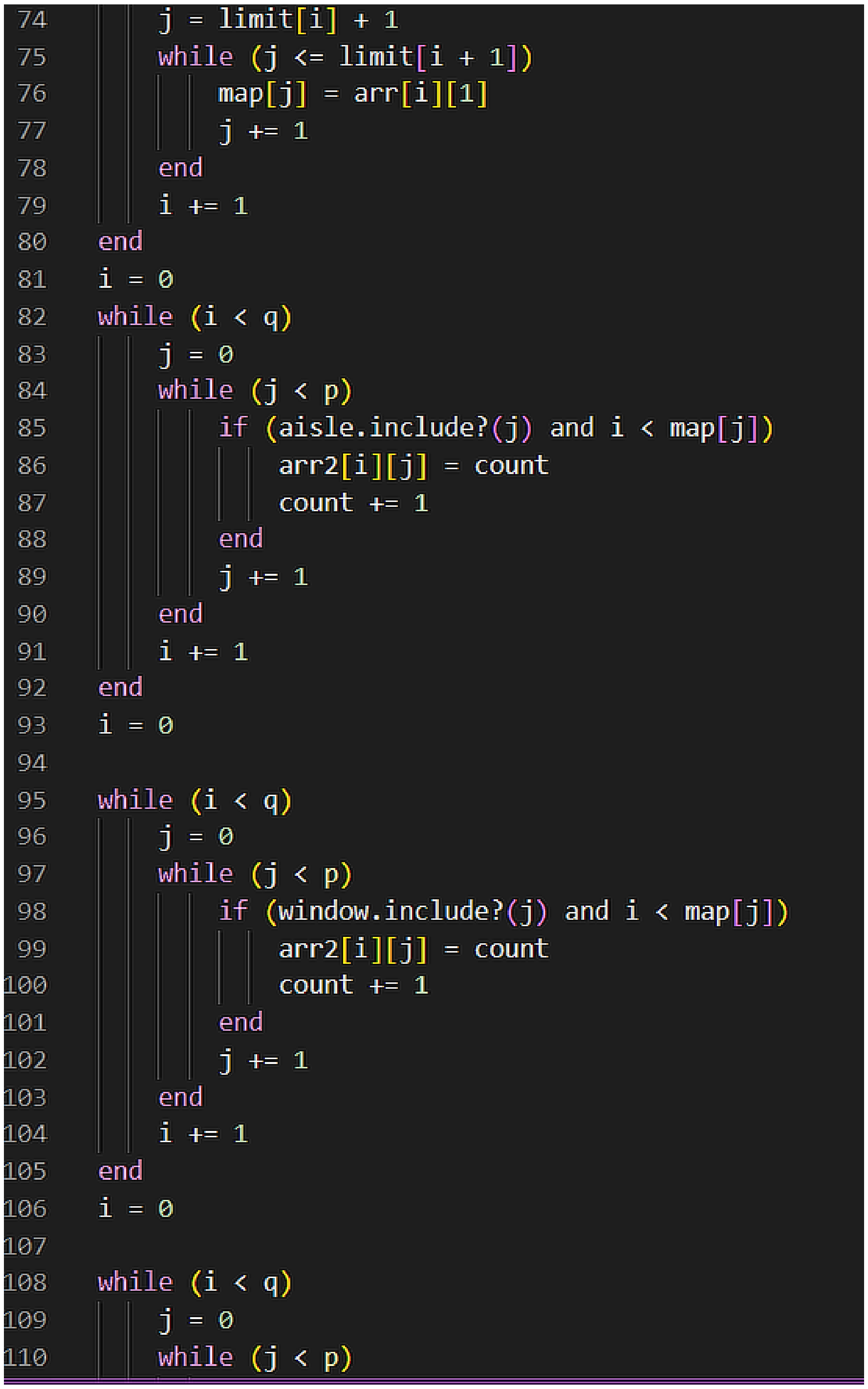
Ruby Code:



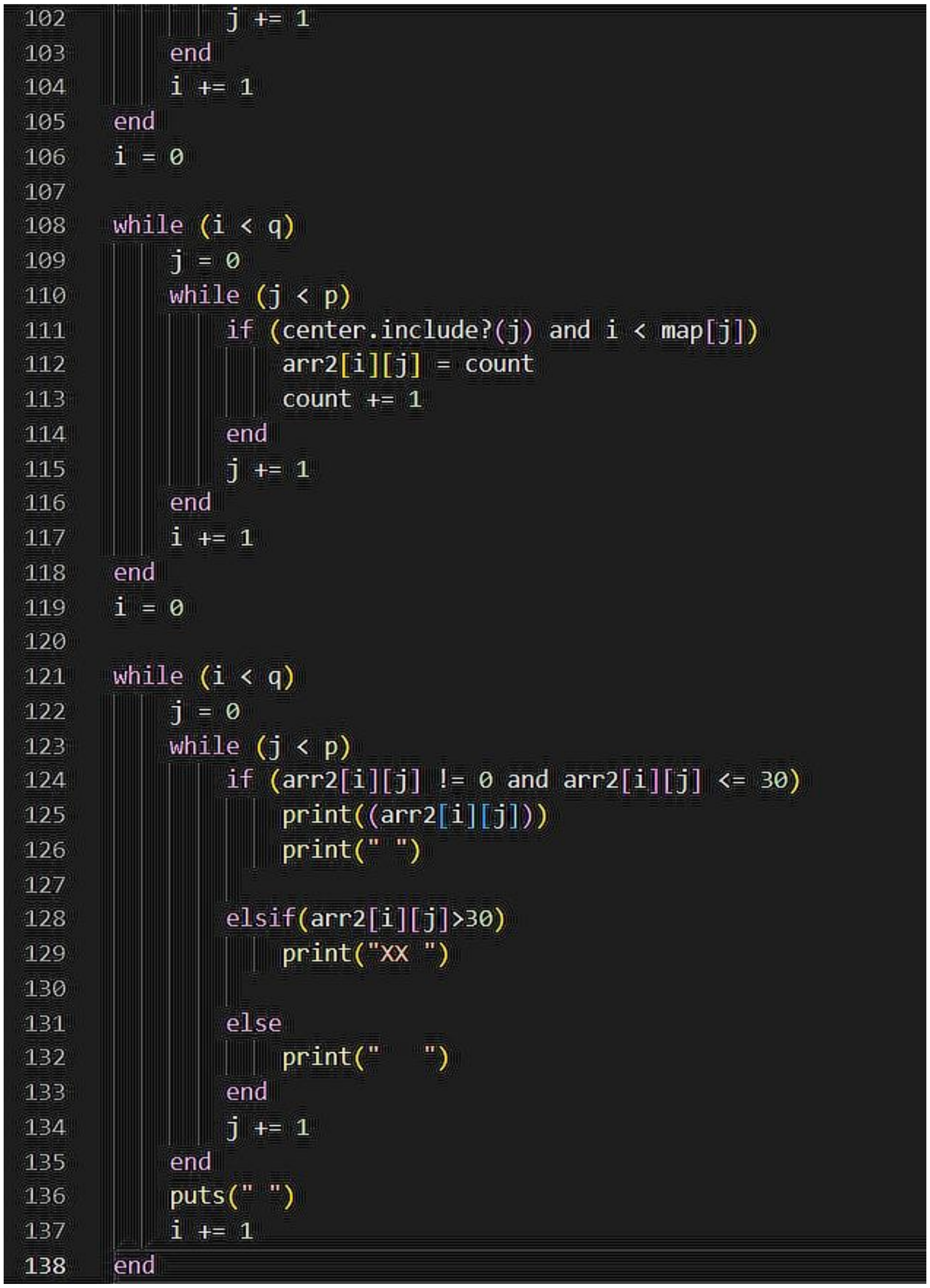
```ruby
arr = [[3, 2], [4, 3], [2, 3], [3, 4]]
n = arr.length
count = 1
k = 30
p = 0
q = -1
aisle =  []
aisle.append(arr[0][0] - 1)
sum = arr[0][0] - 1
i = 1
while (i < n - 1)
    if (arr[i][0] != 2)
        sum += 1
        aisle.append(sum)
        sum += arr[i][0] - 1
        aisle.append(sum)


    else
        sum += 1
        aisle.append(sum)
        sum += 1
        aisle.append(sum)
    end

    i += 1
end
aisle.append(sum + 1)
i = 0
while (i < n)
    p += arr[i][0]
    if (arr[i][1] > q)
        q = arr[i][1]
    end
    i += 1
end
```



```
window =  []
window.append(0)
window.append(p - 1)
center =  []
i = 0

while (i < p)
    if (!aisle.include?(i) and !window.include?(i))
        center.append(i)
    end
    i += 1
end
arr2= Array.new(q) { Array.new(p) { 0 } }
i = 0
while (i < q)
    j = 0
    while (j < p)
        arr2[i][j] = 0
        j += 1
    end
    i += 1
end
sum2 = -1
limit =  []
limit.append(0)
i = 0
while (i < n)
    sum2 += arr[i][0]
    limit.append(sum2)
    i += 1
end
i=0
map = Hash.new
map[0] = arr[0][1]

while (i < n)
    j = limit[i] + 1
```

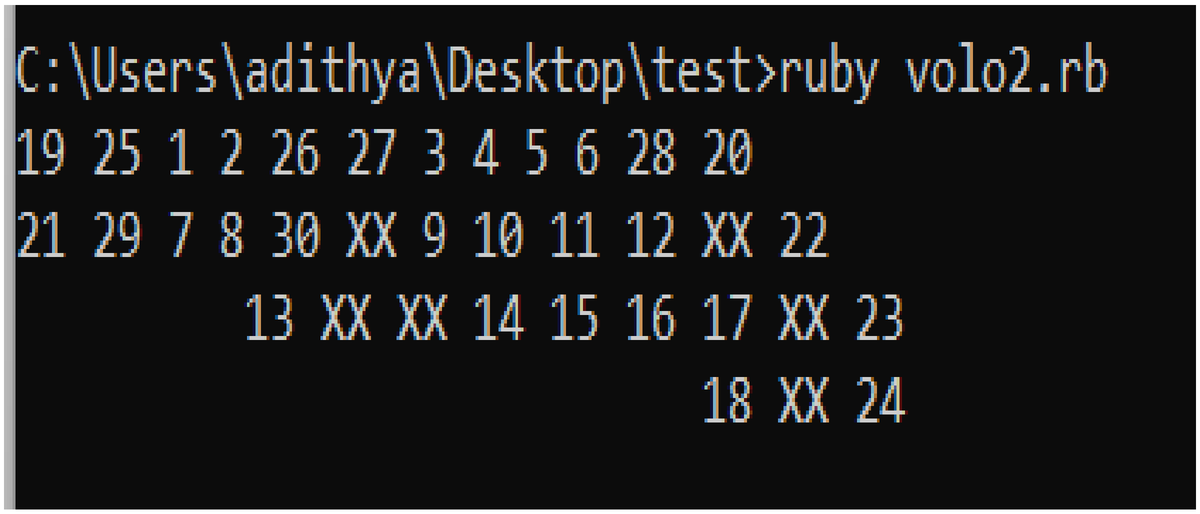


```ruby
    j = limit[i] + 1
    while (j <= limit[i + 1])
        map[j] = arr[i][1]
        j += 1
    end
    i += 1
end
i = 0
while (i < q)
    j = 0
    while (j < p)
        if (aisle.include?(j) and i < map[j])
            arr2[i][j] = count
            count += 1
        end
        j += 1
    end
    i += 1
end
i = 0

while (i < q)
    j = 0
    while (j < p)
        if (window.include?(j) and i < map[j])
            arr2[i][j] = count
            count += 1
        end
        j += 1
    end
    i += 1
end
i = 0

while (i < q)
    j = 0
    while (j < p)
```



```ruby
        j += 1
    end
    i += 1
end
i = 0

while (i < q)
    j = 0
    while (j < p)
        if (center.include?(j) and i < map[j])
            arr2[i][j] = count
            count += 1
        end
        j += 1
    end
    i += 1
end
i = 0

while (i < q)
    j = 0
    while (j < p)
        if (arr2[i][j] != 0 and arr2[i][j] <= 30)
            print((arr2[i][j]))
            print(" ")

        elsif(arr2[i][j]>30)
            print("XX ")

        else
            print("   ")
        end
        j += 1
    end
    puts(" ")
    i += 1
end
```

Output of the Ruby Code:



```
C:\Users\adithya\Desktop\test>ruby volo2.rb
19 25 1 2 26 27 3 4 5 6 28 20
21 29 7 8 30 XX 9 10 11 12 XX 22
         13 XX XX 14 15 16 17 XX 23
                           18 XX 24
```

The test case in which only two columns are present and this leads to both the columns being the aisle seats is also present.