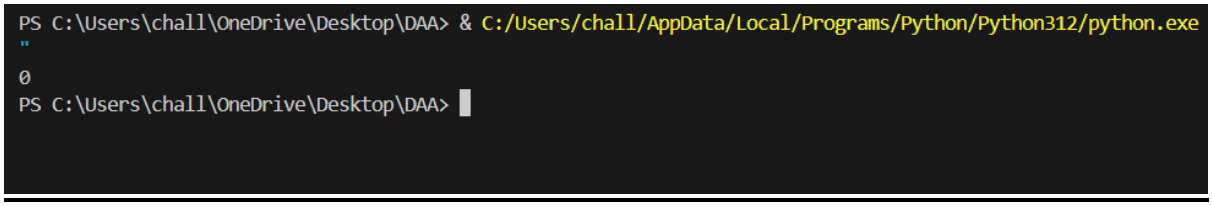
33. Leftmost Column with at Least a One A row-sorted binary matrix means that all elements are 0 or 1 and each row of the matrix is sorted in non-decreasing order. Given a row-sorted binary matrix binaryMatrix, return the index (0-indexed) of the leftmost column with a 1 in it. If such an index does not exist, return -1. You can't access the Binary Matrix directly. You may only access the matrix using a BinaryMatrix interface: ● BinaryMatrix.get(row, col) returns the element of the matrix at index (row, col) (0-indexed). ● BinaryMatrix.dimensions() returns the dimensions of the matrix as a list of 2 elements [rows, cols], which means the matrix is rows x cols. Submissions making more than 1000 calls to BinaryMatrix.get will be judged Wrong Answer. Also, any solutions that attempt to circumvent the judge will result in disqualification. For custom testing purposes, the input will be the entire binary matrix mat. You will not have access to the binary matrix directly.

PROGRAM:

```
class BinaryMatrix:
    def __init__(self, matrix):
        self.matrix = matrix
    def get(self, row: int, col: int) -> int:
        return self.matrix[row][col]
    def dimensions(self) -> list:
        return [len(self.matrix), len(self.matrix[0])]
def leftMostColumnWithOne(binaryMatrix: 'BinaryMatrix') -> int:
    rows, cols = binaryMatrix.dimensions()
    current_row = 0
    current_col = cols - 1
    leftmost_col = -1
    while current_row < rows and current_col >= 0:
        if binaryMatrix.get(current_row, current_col) == 1:
```

```
            leftmost_col = current_col
            current_col -= 1
        else:
            current_row += 1
    return leftmost_col
mat = [[0, 0], [1,1]]
binaryMatrix = BinaryMatrix(mat)
result = leftMostColumnWithOne(binaryMatrix)
print(result)
```

OUTPUT:



TIME COMPLEXITY:

Time complexity for the above code is

F(n)=O(rows+cols)