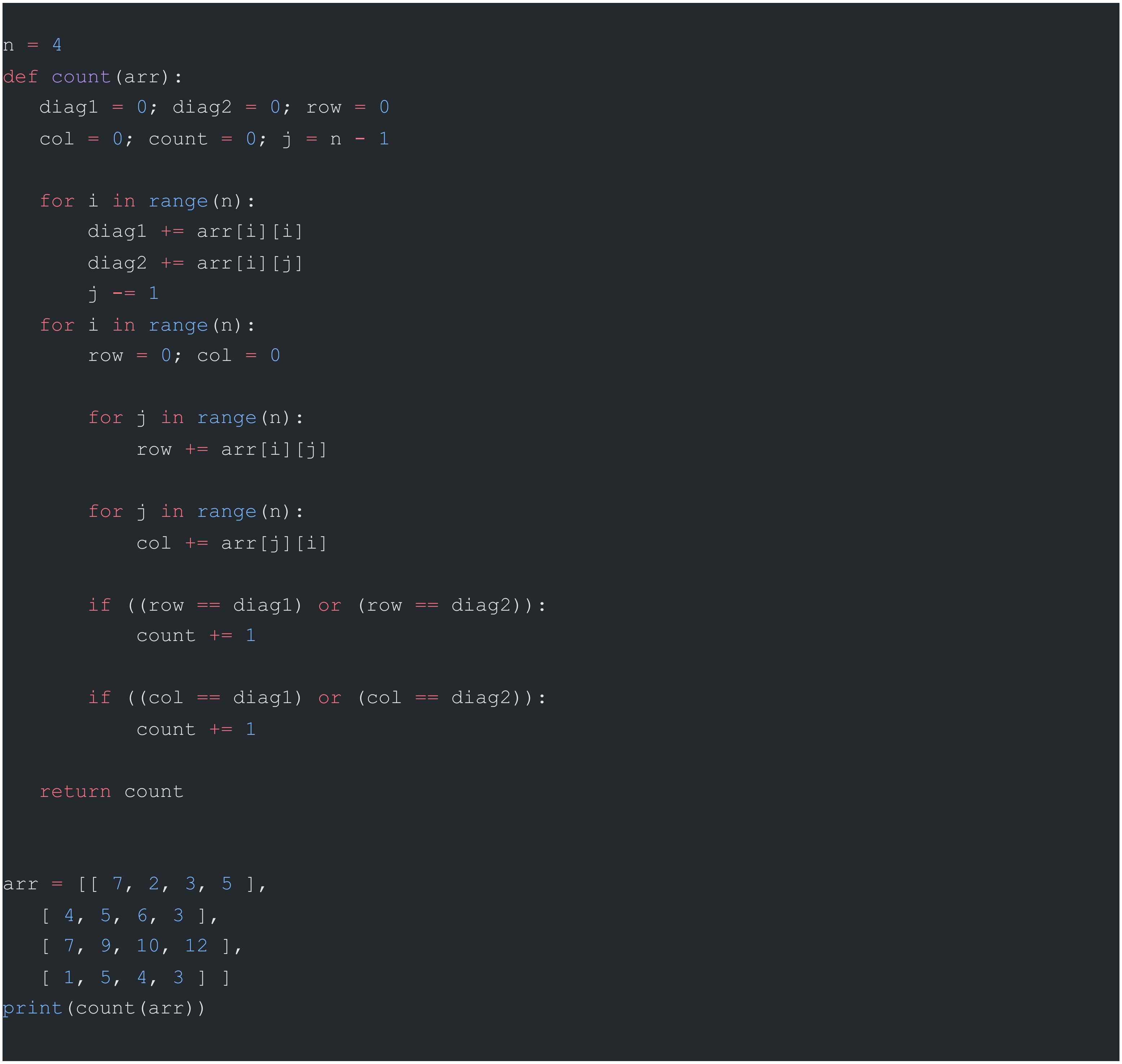
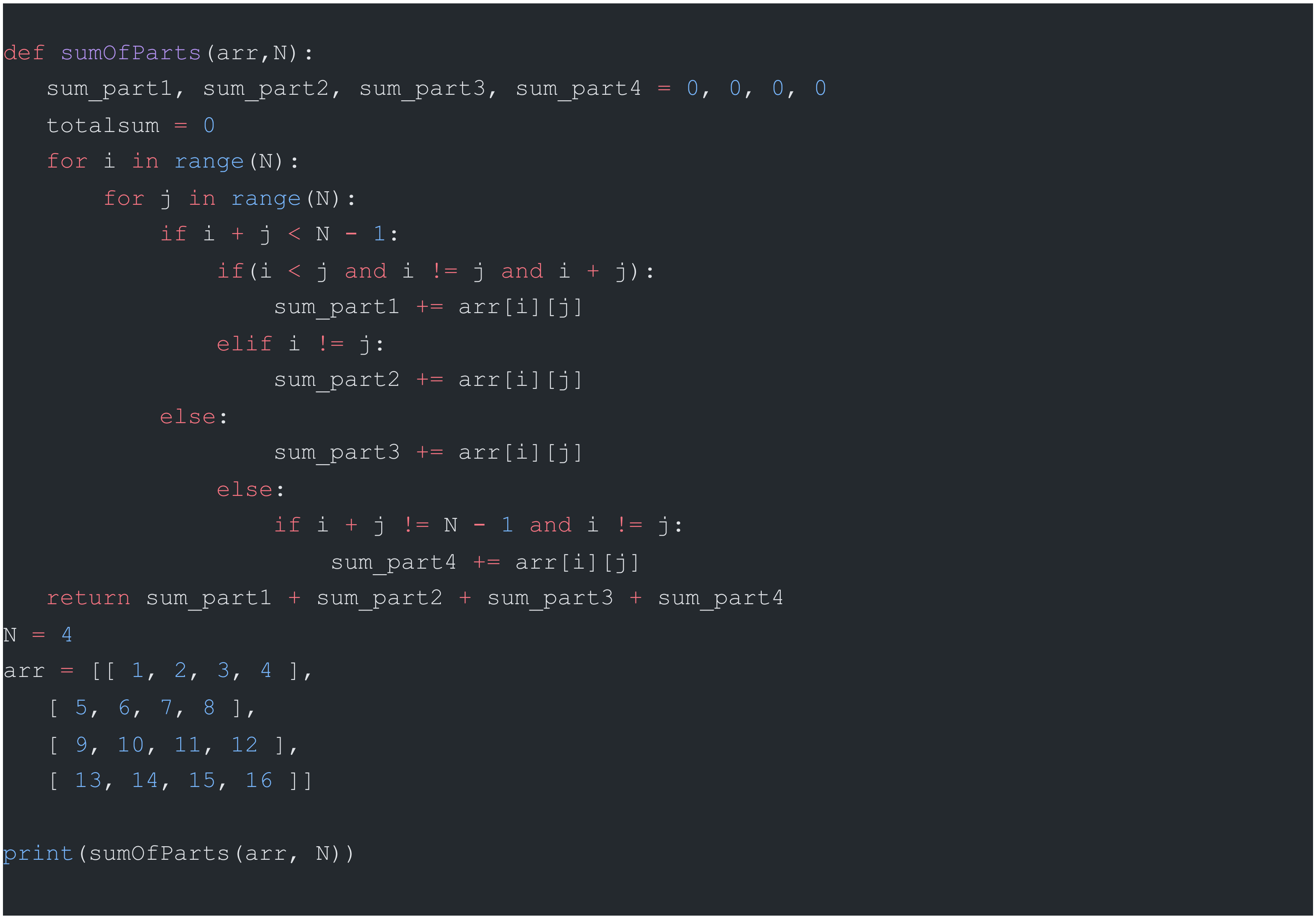
Count rows/columns with sum equals to diagonal sum



```python
n = 4
def count(arr):
    diag1 = 0; diag2 = 0; row = 0
    col = 0; count = 0; j = n - 1

    for i in range(n):
        diag1 += arr[i][i]
        diag2 += arr[i][j]
        j -= 1
    for i in range(n):
        row = 0; col = 0

        for j in range(n):
            row += arr[i][j]

        for j in range(n):
            col += arr[j][i]

        if ((row == diag1) or (row == diag2)):
            count += 1

        if ((col == diag1) or (col == diag2)):
            count += 1

    return count


arr = [[ 7, 2, 3, 5 ],
    [ 4, 5, 6, 3 ],
    [ 7, 9, 10, 12 ],
    [ 1, 5, 4, 3 ] ]
print(count(arr))
```

Sum of non-diagonal parts of a square Matrix



```
def sumOfParts(arr,N):
   sum_part1, sum_part2, sum_part3, sum_part4 = 0, 0, 0, 0
   totalsum = 0
   for i in range(N):
       for j in range(N):
           if i + j < N - 1:
               if(i < j and i != j and i + j):
                   sum_part1 += arr[i][j]
               elif i != j:
                   sum_part2 += arr[i][j]
           else:
                   sum_part3 += arr[i][j]
               else:
                   if i + j != N - 1 and i != j:
                       sum_part4 += arr[i][j]
   return sum_part1 + sum_part2 + sum_part3 + sum_part4
N = 4
arr = [[ 1, 2, 3, 4 ],
   [ 5, 6, 7, 8 ],
   [ 9, 10, 11, 12 ],
   [ 13, 14, 15, 16 ]]

print(sumOfParts(arr, N))
```