Karibay Sanzhar

SE-2017

1.

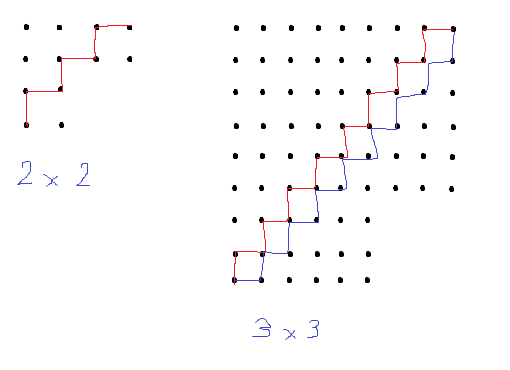
# factorial function  
def factorial(N):  
 result = 1;  
  
 while (N > 0):  
 result = result \* N;  
 N -= 1;  
  
 return result;  
  
  
# I decided not to make the algorithm to go thru every possible way, but to count it!   
def countWays(N):  
 # the formula of every possible way to get from A to B  
 total = factorial(N + N);  
 index = factorial(N);  
  
 return (total // index) // index;  
  
  
N = 4;  
print("Ways =", countWays(4));

2.

# factorial function  
def factorial(N):  
 result = 1;  
  
 while (N > 0):  
 result = result \* N;  
 N -= 1;  
  
 return result;  
  
  
# I decided not to make the algorithm to go thru every possible way, but to count it!   
def countWays(N):  
 # the formula of every possible way to get from A to B  
 total = factorial(N + N);  
 index = factorial(N);  
  
 return (total // index) // index // 2;  
  
  
N = 4;  
print("Ways =", countWays(4));

3.

def foo(N):  
 if N % 2 == 0:  
 print("1 Way")  
 return  
 print("2 Ways")



4.

def foo(space, meteor):  
 # Initial coordinates  
 arr = [0, 0]  
 # counter of steps  
 counter = 0  
  
 for i in range(max(space)):  
 # if meteor is on the direction of the shape, then Y then we just go up or right (depends on minimum coordinate),   
 # which means 2 steps  
 if arr[0] == meteor[0] and arr[1] == meteor[1]:  
 arr[arr.index(min(arr))] += 2  
 counter += 2  
 # X direction 1 step  
 if arr[0] < space[0]:  
 arr[0] += 1  
  
 # Y direction 1 step  
 if arr[1] < space[1]:  
 arr[1] += 1  
  
 counter += 1  
 return counter, arr

5.

def foo(arr):  
 # Initial coordinates  
 arr2 = [0, 0, 0]  
 # counter of steps  
 counter = 0  
  
 for i in range(max(arr)):  
 # X direction 1 step  
 if arr2[0] < arr[0]:  
 arr2[0] += 1  
  
 # Y direction 1 step  
 if arr2[1] < arr[1]:  
 arr2[1] += 1  
  
 # Z direction 1 step  
 if arr2[2] < arr[2]:  
 arr2[2] += 1  
  
 counter += 1  
 return counter, arr2