'''At the annual meeting of Board of Directors of Acme Inc. If everyone attending shakes hands exactly one time with every other attendee, how many handshakes are there?

Example

There are attendees, , and . shakes hands with and , and shakes hands with . Now they have all shaken hands after handshakes.

Function Description

Complete the handshakes function in the editor below.

handshakes has the following parameter:

int n: the number of attendees

Returns

int: the number of handshakes

Input Format

The first line contains the number of test cases .

Each of the following lines contains an integer, .

Constraints

Sample Input

2

1

2

Sample Output

0

1'''

import math

import os

import random

import re

import sys

#

# Complete the 'handshake' function below.

#

# The function is expected to return an INTEGER.

# The function accepts INTEGER n as parameter.

#

def handshake(n):

# Write your code here

return n\*(n-1)//2

if \_\_name\_\_ == '\_\_main\_\_':

fptr = open(os.environ['OUTPUT\_PATH'], 'w')

t = int(input().strip())

for t\_itr in range(t):

n = int(input().strip())

result = handshake(n)

fptr.write(str(result) + '\n')

fptr.close()

'''Luke is daydreaming in Math class. He has a sheet of graph paper with rows and columns, and he imagines that there is an army base in each cell for a total of bases. He wants to drop supplies at strategic points on the sheet, marking each drop point with a red dot. If a base contains at least one package inside or on top of its border fence, then it's considered to be supplied. For example:

image

Given and , what's the minimum number of packages that Luke must drop to supply all of his bases?

Example

Packages can be dropped at the corner between cells (0, 0), (0, 1), (1, 0) and (1, 1) to supply bases. Another package can be dropped at a border between (0, 2) and (1, 2). This supplies all bases using packages.

Function Description

Complete the gameWithCells function in the editor below.

gameWithCells has the following parameters:

int n: the number of rows in the game

int m: the number of columns in the game

Returns

int: the minimum number of packages required

Input Format

Two space-separated integers describing the respective values of and .

Constraints

Sample Input 0

2 2

Sample Output 0

1'''

import math

import os

import random

import re

import sys

#

# Complete the 'gameWithCells' function below.

#

# The function is expected to return an INTEGER.

# The function accepts following parameters:

# 1. INTEGER n

# 2. INTEGER m

#

def gameWithCells(n, m):

# Write your code here

return math.ceil(n/2) \* math.ceil(m/2)

if \_\_name\_\_ == '\_\_main\_\_':

fptr = open(os.environ['OUTPUT\_PATH'], 'w')

first\_multiple\_input = input().rstrip().split()

n = int(first\_multiple\_input[0])

m = int(first\_multiple\_input[1])

result = gameWithCells(n, m)

fptr.write(str(result) + '\n')

fptr.close()