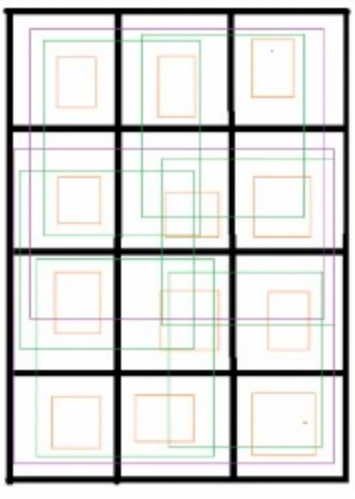
Count number of squares in a rectangle

Given a m x n rectangle, how many squares are there in it?

Examples:

```
Input:  m = 2, n = 2
Output: 5
There are 4 squares of size 1x1 +
          1 square of size 2x2.

Input: m = 4, n = 3
Output: 20
There are 12 squares of size 1x1 +
          6 squares of size 2x2 +
          2 squares of size 3x3.
```



We strongly recommend you to minimize your browser and try this yourself first.

Let us first solve this problem for m = n, i.e., for a square:

For m = n = 1, output: 1

For m = n = 2, output: 4 + 1 [4 of size 1×1 + 1 of size 2×2]

For m = n = 3, output: 9 + 4 + 1 [4 of size 1×1 + 4 of size 2×2 + 1 of size 3×3]

For m = n = 4, output 16 + 9 + 4 + 1 [16 of size 1×1 + 9 of size 2×2 + 4 of size 3×3 + 1 of size 4×4]

In general, it seems to be n^2 + (n-1)^2 + ... 1 = n(n+1)(2n+1)/6

Let us solve this problem when m may not be equal to n:

Let us assume that m <= n

From above explanation, we know that number of squares in a m x m matrix is m(m+1)(2m+1)/6

What happens when we add a column, i.e., what is the number of squares in m x (m+1) matrix?

When we add a column, number of squares increased is m + (m-1) + ... + 3 + 2 + 1
[m squares of size 1×1 + (m-1) squares of size 2×2 + ... + 1 square of size m x m]

Which is equal to m(m+1)/2

So when we add (n-m) columns, total number of squares increased is (n-m)*m(m+1)/2.

So total number of squares is m(m+1)(2m+1)/6 + (n-m)*m(m+1)/2.

Using same logic we can prove when n <= m. So, in general,

```
Total number of squares = m x (m+1) x (2m+1)/6 +
                          (n-m) x m x (m+1)/2

when n is larger dimension
```

.

Using above logic for rectangle, we can also prove that number of squares in a square is n(n+1)(2n+1)/6

Below is C++ implementation of above formula.

```
// C++ program to count squares in a rectangle of
// size m x n
#include<iostream>
using namespace std;

// Returns count of all squares in a rectangle
// of size m x n
int countSquares(int m, int n)
{
   // If n is smaller, swap m and n
   if (n < m)
      swap(m, n);

   // Now n is greater dimension, apply formula
   return  m*(m+1)*(2*m+1)/6 + (n-m)*m*(m+1)/2;
}

// Driver method
int main()
{
   int m = 4, n = 3;
   cout << "Count of squares is " << countSquares(m, n);
}
```

Run on IDE

Count of Squares is 20