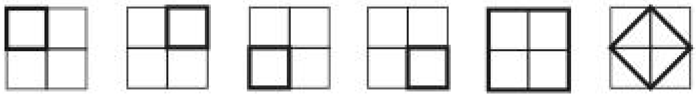
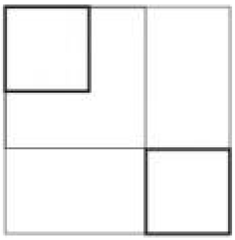
Cho một lưới có điểm lưới tọa độ nguyên kích thước $N \times N$. Hãy xác định số hình vuông có thể tạo được sao cho 4 đỉnh của hình vuông là 4 điểm lưới.

Ví dụ: Lưới 3×3 ta có thể xác định được 6 hình vuông.

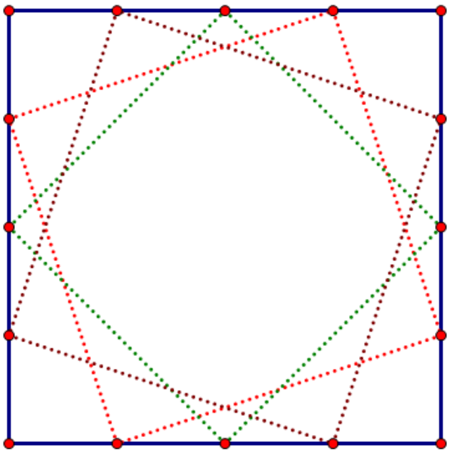


Lời giải:

- Một hình vuông với K điểm ở trên một cạnh và các cạnh song song với trục tọa độ có thể được di chuyển theo $(N - K + 1) * (N - K + 1)$ vị trí khác nhau trên lưới.



- Chúng ta xét trường hợp số hình vuông có các cạnh song song với các đường chéo. Dễ dàng thấy được nếu lưới vuông có K điểm mỗi cạnh thì sẽ có $K - 2$ hình vuông có cạnh song song với các đường chéo.



Từ đó ta có công thức tổng quát cho trường hợp $N * N$ như sau:

$$= \sum_{K=2}^{N} (N - K + 1) \times (N - K + 1) + \sum_{K=2}^{N} (N - K + 1) \times (N - K + 1) \times (K - 2)$$

$$= \sum_{K=2}^{N} (N - K + 1) \times (N - K + 1) \times (K - 1)$$

$$= \sum_{K=1}^{N-1} (N-K) \times (N-K) \times K$$

$$= \sum_{K=1}^{N-1} K \times N^2 - 2K^2 \times N + K^3$$

$$= N^2 \sum_{K=1}^{N-1} K - 2 \times N \sum_{K=1}^{N-1} K^2 + \sum_{K=1}^{N-1} K^3$$

$$= N^2 \frac{N(N-1)}{2} - 2N * \frac{N(N-1)(2N-1)}{6} + \left[\frac{N(N-1)}{2}\right]^2$$

ở đây sử dụng các công thức:

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2} \quad \sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6} \quad \sum_{k=1}^{n} k^3 = \left[\frac{n(n+1)}{2}\right]^2$$

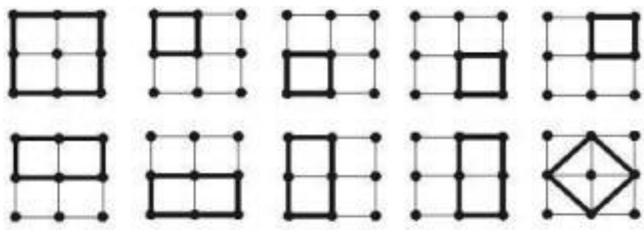
Cho lưới $n \times m$. Xác định số lượng hình chữ nhật có các đỉnh là điểm nguyên trên lưới.

Dữ liệu: Vào từ file văn bản **RECTANGLES.INP** gồm một dòng duy nhất chứa 2 số nguyên dương m, n $(2 \le m, n \le 400)$.

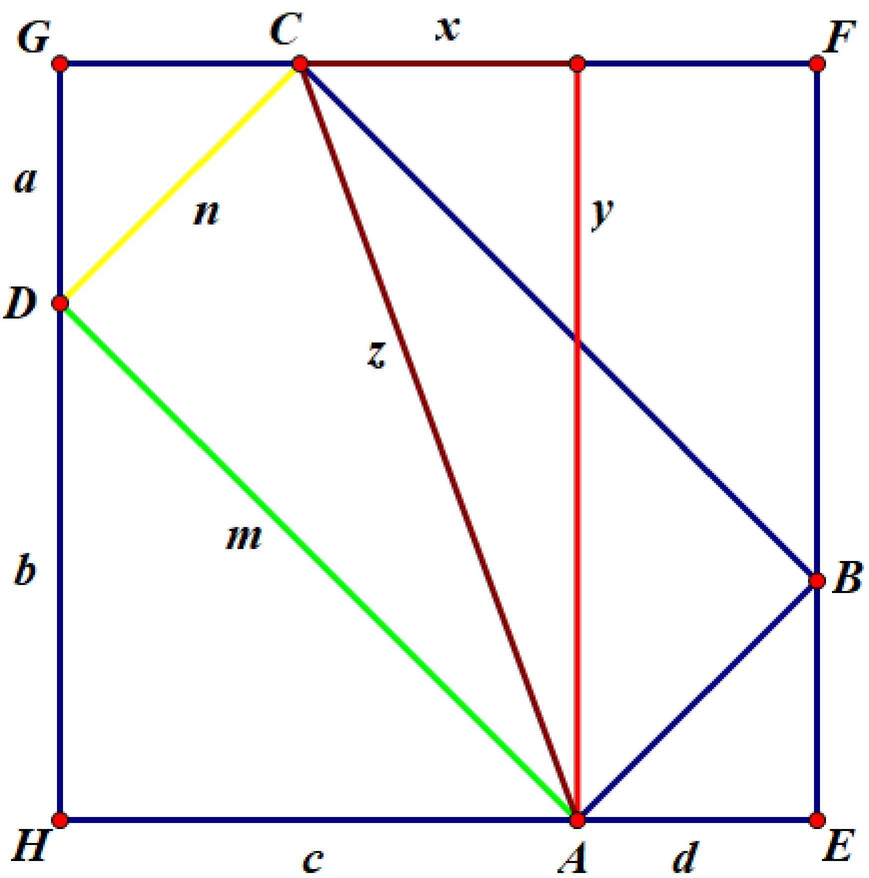
Kết quả: Ghi ra file văn bản **RECTANGLES.INP** một số nguyên duy nhất là số lượng hình chữ nhật đếm được.

Ví dụ:

RECTANGLES.INP	RECTANGLES.INP
3 3	10



Lời giải:



Chúng ta đếm số hình chữ nhật tạo thành trong hình chữ nhật cỡ $H * W$. Giả sử hình chữ nhật mới tạo thành chia 2 cạnh của hình chữ nhật cỡ $H * W$ thành các đoạn độ dài a, b và c, d. Sử dụng định lý Pythagore chúng ta có:

- $a^2 + d^2 = n^2$
- $b^2 + c^2 = m^2$
- $n^2 + m^2 = z^2$
- $(A + B)^2 = y^2$
- $(C - D)^2 = x^2$
- $x^2 + y^2 = z^2$

Từ đó chúng ta có:

$$(a + b)^2 + (c - d)^2 = a^2 + b^2 + c^2 + d^2$$

$$\Leftrightarrow ab = cd$$

$$\Leftrightarrow c^2 - Wc + a(H - a) = 0 \text{ (vì } b = H - a, d = W - c)$$

Từ đó, cố định a, chúng ta sẽ tìm được c $(0 \leq c \leq W)$

Do đó, ta mất $O(H)$ để đếm số lượng hình chữ nhật trong một hình chữ nhật có kích thước $H * W$. Hình chữ nhật này có thể được đặt trên $(M - W + 1) * (N - H + 1)$ vị trí trên lưới kích thước $N * M$. Vì vậy độ phức tạp là $O(N * M^2)$

```
#include <bits/stdc++.h>
using namespace std;
fstream in ( "RECTANGLES.INP" , ios::in  );
fstream out( "RECTANGLES.OUT", ios::out );
const int NMAX = 4e3 + 5;
int sqr[NMAX * NMAX];
int main( void ) {
    ios::sync_with_stdio( false );
    int n, m;
    in >> n >> m;
    fill( sqr, sqr + m * m, -1 );
    for( int i = 0; i < m; i ++ )
        sqr[i * i] = i;
    long long ans = 0;
    for( int i = 1; i < n; i ++ ) {
        for( int j = 1; j < m; j ++ ) {
            int aux = 1;
            for( int k = 1; k < i; k ++ ) {
                int dlt = j * j - 4 * k * ( i - k );
                if( dlt < 0 || sqr[dlt] < 0 )
                    continue;
                int sl1 = ( j + sqr[dlt] ) / 2;
                int sl2 = ( j - sqr[dlt] ) / 2;
                if( ( j + sqr[dlt] ) % 2 == 0 && 1 <= sl1 && sl1 < j )
                    aux ++;
```

```
                if( ( j + sqr[dlt] ) % 2 == 0 && 1 <= sl2 && sl2 < j &&
dlt != 0 )
                    aux ++;
            }
            ans += aux * 1LL * ( n - i ) * ( m - j );
        }
    }
    out << ans;
    return 0;
}
```