**Sum and Multiply**

12194% *of* 4017 *of* 157[Almosawi](https://www.codewars.com/users/Almosawi)

C#

* [Train Again](https://www.codewars.com/kata/sum-and-multiply/train/csharp)
* [Next Kata](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/sum-and-multiply/solutions/csharp)

[Forks (1)](https://www.codewars.com/kata/sum-and-multiply/forks/csharp)

[Discourse (16)](https://www.codewars.com/kata/sum-and-multiply/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Write a function that accepts two parameters (sum and multiply) and find two numbers [x, y], where x + y = sum and x \* y = multiply.

Example:

sum = 12 and multiply = 32

In this case, x equals 4 and y equals 8.

x = 4

y = 8

Because

x + y = 4 + 8 = 12 = sum

x \* y = 4 \* 8 = 32 = multiply

The result should be [4, 8].

Note:

0 <= x <= 1000

0 <= y <= 1000

If there is no solution, your function should return null (or None in python).

You should return an array (list in python) containing the two values [x, y] and it should be sorted in ascending order.

One last thing: x and y are integers (no decimals).

[**https://www.codewars.com/kata/sum-and-multiply/csharp**](https://www.codewars.com/kata/sum-and-multiply/csharp)

**** [**Souzooka**](https://www.codewars.com/users/Souzooka)

public class Kata

{

public static int[] SumAndMultiply(int sum, int multiply)

{

for (int i = 0; i <= sum / 2; ++i)

{

if ((sum - i) \* i == multiply) { return new int[] {i, sum - i}; }

}

return null;

}

}

* + Best Practices0
  + Clever1
* 0
* [Fork](https://www.codewars.com/kumite/new?group_id=599d74525c795f4381001bb5&review_id=599d74495c795f641a001bb1)
* [Link](https://www.codewars.com/kata/reviews/599d74495c795f641a001bb1/groups/599d74525c795f4381001bb5)

**** [**Egorka11**](https://www.codewars.com/users/Egorka11)

using System;

public class Kata {

public static int[] SumAndMultiply(int s, int m) {

int x = 0, y = 0;

if(s \* s >= 4 \* m && (s - Math.Sqrt(s \* s - 4 \* m)) % 2 == 0)

x = (s - (int)Math.Sqrt(s \* s - 4 \* m)) / 2;

if(s \* s >= 4 \* m && (s + Math.Sqrt(s \* s - 4 \* m)) % 2 == 0)

y = (s + (int)Math.Sqrt(s \* s - 4 \* m)) / 2;

return x == 0 && y == 0 ? null : new int[]{x, y};

}

}