Uma abordagem geométrica / aritmética

No primeiro bit de código, verificamos se um número era divisível por 4 e / ou 5 e, dessa forma, verificamos cada número apenas uma vez. Outra solução seria encontrar a soma de todos os números divisíveis por três e a soma de todos os números divisíveis por 5. poderíamos definir uma função que disesse. Onde **n** é o divisor e **p** é o maior número que queremos verificar . Que neste caso p = 999, en = {3,5} neste caso temos o número do contador que são divisíveis por 3 e 5 duas vezes, e, portanto, precisamos subtraí-los de forma que a solução seria

No entanto, precisamos colocar algo inteligente dentro da função sumDivisibleBy, caso contrário, teremos uma solução ainda mais lenta

private int SumDivisbleBy (int n, int p) {

}

SumDivisibleBy(3,999) + SumDivisibleBy(5,999) - SumDivisibleBy(15,999).

Então, vamos olhar para os números divisíveis por n = 3:

3 + 6 + 9 + 12 + ... + 999 ... = 3 \* (1 + 2 + 3 + 4 + ... + 333) E os números divisíveis por n = 5

5 + 10 + 15 + 20 + ... + 995 = 5 \* (1 + 2 + 3 + 4 + ... + 199) O truque é expressar a soma por outros meios, e neste caso a soma 1+ 2 + 3 + 4 + ... + N = (N \* (N + 1) / 2).

Uma explicação geométrica é fornecida aqui e uma explicação aritmética é fornecida aqui. Uma explicação mais geral sobre a progressão aritmétrica é fornecida na Wikipedia.

Assim, a sequência para qualquer número divisível por n pode ser escrita como

n \* N \* (N + 1) / 2.

N é o número mais alto menor que p divisível por n. No caso de 5, isso é 995.

no entanto, com divisão inteira N pode ser expresso como N = p / n