

PART 1.6

1. Escreva um programa que recebe como entradas (utilize a função `input`) dois números inteiros correspondentes à largura e à altura de um retângulo, respectivamente. O programa deve imprimir, usando repetições encaixadas (laços aninhados), uma cadeia de caracteres que represente o retângulo informado com caracteres " na saída.
2. Refaça o exercício 1 imprimindo os retângulos sem preenchimento, de forma que os caracteres que não estiverem na borda do retângulo sejam espaços.
3. Escreva a função `n_primos` que recebe como argumento um número inteiro maior ou igual a 2 como parâmetro e devolve a quantidade de números primos que existem entre 2 e `n` (incluindo 2 e, se for o caso, `n`).
4. Dizemos que um número h é uma **hipotenusa** de um triângulo inteiro se existe um triângulo retângulo com lados inteiros cuja hipotenusa é igual a esse número. Em outras palavras, h é uma hipotenusa se existem números inteiros i e j tais que:

$$h^2 = i^2 + j^2 \tag{1}$$

Escreva uma função `soma_hipotenusas` que receba como parâmetro um número inteiro positivo `n` e devolva a soma de todos os inteiros entre 1 e `n` que são comprimento da hipotenusa de algum triângulo retângulo com catetos inteiros.

Um mesmo número pode ser hipotenusa de vários triângulos, mas deve ser somado apenas uma vez. Uma boa solução para este exercício é fazer um laço de 1 até `n` testando se o número é hipotenusa de algum triângulo e somando em caso afirmativo. Uma solução que dificilmente vai dar certo é fazer um laço construindo triângulos e somando as hipotenusas inteiras encontradas.

Primeiro faça uma função `é_hipotenusa` que diz se um número inteiro é o comprimento da hipotenusa de um triângulo com lados de comprimento inteiro ou não.