

divmod

Descrição

Muitas linguagens de programação antigas implementavam uma função chamada `divmod` (abreviação de *divisão e módulo*), que retorna o quociente e o restante ao dividir dois números usando a divisão inteira. Na verdade, muitas linguagens modernas também implementam isso, uma vez que acaba sendo útil em várias situações.

As primeiras duas linhas em seu programa devem configurar duas variáveis `dividend` e `divisor` (uma para representar o dividendo e outra para representar o divisor). Seu programa deve atribuir uma tupla de formato (*quociente, resto*) a uma variável chamada `out`.

Por exemplo, se o seu programa começou com:

```
dividend = 7
divisor = 2
```

Em seguida, então `out` deverá ser `(3, 1)`, já que dividir 7 por 2 dá quociente 3 com resto 1.

Você também deve se certificar de que seu programa funcione independentemente de `dividend` e `divisor` serem floats ou inteiros, mas ele só precisa funcionar para *números positivos*. Para os nossos propósitos neste programa, o tipo de valor que você coloca em `out` não importa, desde que o valor esteja correto.

Seria fácil implementar tal programa usando os operadores `//` e `%` integrados do Python. Mas não estamos interessados em escolher o caminho mais fácil! Como tal, sua definição não deve usar os operadores `//`, `/` ou `%`, nem a função integrada `divmod` (sim, Python é uma das linguagens que considerou essa função importante o suficiente para incluí-la).

Notas

Nosso verificador Python é específico sobre nomes de variáveis e, portanto, é importante que você chame as variáveis mencionadas acima de `dividend`, `divisor` e `out`, em vez de dar-lhes nomes diferentes. Variáveis diferentes daquela podem ser chamadas de como você quiser.

Testando

Certifique-se de escrever e testar seu código *manualmente* antes de usar Python. Desenhe diagramas de ambiente para alguns casos de teste diferentes. Então, somente quando você estiver convencido de que funcionará, execute seu código com Python em sua própria máquina como um teste.

Submissão

Quando estiver pronto, faça upload do seu arquivo Python no Problema **1.3** no Gradescope. Lembre de nomear seu arquivo `p1_3.py`.