

Exercício 1

Escreva um programa em Python que leia, sucessivamente, uma string representando um inteiro maior ou igual a 2, e indique se o inteiro é um número primo.

Da primeira vez que for introduzido um inteiro n não primo, o programa imprime " n não é primo", depois imprime o total de primos introduzidos, e termina.

Observe os exemplos de interacção abaixo, que ilustram o comportamento do programa incluindo os formatos de input e output. Considere que os exemplos fazem parte do contrato do programa.

Restrições: não pode ocorrer no programa nenhuma das sequências de caracteres nesta lista: ["for", "def", "import", "break", "continue"]

Nos exemplos, a cor preta denota input, e a cor azul denota output. Estas são as cores obtidas com a shell do Idle numa instalação típica com configurações habituais. As cores podem variar em instalações locais específicas.

Exemplo A

```
17
17 é primo
2
2 é primo
7
7 é primo
20
20 não é primo
Introduziu 3 primos
```

Exemplo B

```
16
16 não é primo
Introduziu 0 primos
```

Exemplo C

```
11
11 é primo
16
16 não é primo
Introduziu 1 primos
```

Exercício 2



As traças são irremediavelmente atraídas pela luz. Simule o movimento de uma traça que é atraída para a luz num espaço discreto unidimensional em que a traça começa sempre à esquerda da luz. Essa traça começa com velocidade 0 e progressivamente, de dois em dois instantes, aumenta a velocidade em 1 unidade. O movimento de uma traça com velocidade 3, por exemplo, significa que ela salta da coordenada n para a coordenada $n+3$ entre instantes consecutivos. Considere que a traça nunca ultrapassa a luz, independentemente da velocidade. Ou seja, acaba sempre na mesma posição da luz, mesmo que tivesse velocidade suficiente para a ultrapassar num dado instante.

Escreva um programa em Python em que o utilizador introduza a posição da traça e da luz, por esta ordem, e em que o programa simule a evolução do movimento da traça, instante após instante, até embater na luz, tendo que imprimir as suas posições sucessivas. *Deve imprimir também a posição inicial, antes de efectuar qualquer movimento.*

Restrições: não pode ocorrer no programa nenhuma das sequências de caracteres nesta lista: ["for", "def", "import", "break", "continue"]

Output

For example:

Test	Input	Result
#teste 1	8 12	8
		8
		8
		9
		10
		12
#teste 2	8 11	8
		8
		8
		9
		10
		11
#teste 3	-5 13	-5
		-5
		-5
		-4
		-3
		-1
		1
		4
		7
		11
		13

Exercício 3



Na descrição seguinte, *rapidez* é o valor absoluto da velocidade.

Num espaço discreto unidimensional limitado por duas paredes, entre $-dim$ e dim , há um carrinho de choques que arranca com uma determinada velocidade inicial, positiva. Seguindo a convenção habitual de representar os números inteiros (que aqui são posições) ordenadamente num eixo orientado da esquerda para a direita, diremos que o carrinho arranca sempre para a direita. Se a velocidade for de $+2$, isso quer dizer que, entre instantes consecutivos, o carrinho salta de 2 em 2 posições da esquerda para a direita. Noutro exemplo, se a velocidade for de -3 , isso quer dizer que, entre instantes consecutivos, o carrinho salta de 3 em 3 posições da direita para a esquerda. Quando o carrinho bate numa parede, independentemente da sua velocidade, fica nessa parede, inverte o sentido e perde uma certa quantidade de rapidez, continuando depois o seu movimento até bater na parede oposta, inverter o sentido de novo, perder rapidez e continuar o zigzag enquanto tiver velocidade diferente de 0. Após um determinado número de instantes o carro acabará inevitavelmente por parar. Por exemplo, se a velocidade inicial for de $+5$ e a perda de rapidez for de 1 unidade em cada choque, irá chocar 5 vezes com as paredes, sendo 3 vezes com a da direita e 2 com a da esquerda, no seu movimento em zigzag, acabando por parar na parede direita.

Escreva um programa em Python que leia do utilizador, por esta ordem, o valor de dim que determina a dimensão do espaço, a posição inicial do carro de choques, a sua velocidade inicial e a perda de rapidez quando choca com uma parede, imprimindo as coordenadas sucessivas do carro até parar ($rapidez = 0$). O valor dim tem de ser maior do que zero. A posição inicial do carro de choques tem de ser maior do que $-dim$ e menor do que dim . A velocidade inicial e a perda de rapidez têm ambas de ser maiores do que 0.

Note que a rapidez nunca pode ser negativa. A velocidade alterna de sinal em cada colisão com uma parede, excepto na última (em que o carro pára).

O valor da coordenada inicial do carro é impresso no output.

Restrições: não pode ocorrer no programa nenhuma das sequências de caracteres nesta lista: ["for", "def", "import", "break", "continue"]

Output:

For example:

Test	Input	Result
# teste 1	10	-3
	-3	3
	6	9
	2	10
		6
		2
		-2
		-6
		-10
		-8
		-6
		-4
		-2
		0
		2
		4
		6
		8
		10
# teste 2	10	-3
	-3	5
	8	10
	3	5
		0
		-5
		-10
		-8
		-6
		-4
		-2
		0
		2
		4
		6
		8
		10