

Iniciado em	sexta, 27 ago 2021, 18:49
Estado	Finalizada
Concluída em	domingo, 12 set 2021, 20:38
Tempo empregado	16 dias 1 hora
Avaliar	10,00 de um máximo de 10,00(100%)

Aviso!

?

Questão 1

Correto

Atingiu 1,00 de 1,00

Recursão com Resultado

Crie um [programa](#) que lê um número inteiro a ($2 \leq a$), chama a [função](#) *maiorQue1000*, que tem como argumento o número inteiro a , e imprime o valor retornado da [função](#) *maiorQue1000*.

A [função](#) *maiorQue1000* checa se a é maior que 1000. Se for, retorna a . Se não, chama recursivamente *maiorQue1000* com a elevado ao quadrado como argumento e retorna o resultado dessa chamada recursiva.

Dessa forma, a [função](#) *maiorQue1000* se chamará recursivamente até que a seja maior que 1000.

Entrada

A entrada contém um número inteiro.

Saída

A saída consiste na impressão de um número maior que 1000.

Observações

- No primeiro exemplo de teste, foi digitado 2, retornando 65536.
- No segundo exemplo de teste, foi digitado 1001, retornando 1001.
- No terceiro exemplo de teste, foi digitado 5, retornando 390625.

Particularidade do Tópico

- Atenção, a criação de uma [função](#) com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da [função](#)).

For example:

Input	Result
2	65536
1001	1001
5	390625

Input	Result
5	390625

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 a = int(input())
2
3 def maiorQue1000(a):
4     if a > 1000:
5         return a
6     else:
7         return maiorQue1000(a**2)
8
9
10 print(maiorQue1000(a))
11
12
13

```

	Input	Expected	Got	
✓	2	65536	65536	✓
✓	1001	1001	1001	✓
✓	5	390625	390625	✓
✓	10	10000	10000	✓

Aviso!

?

	Input	Expected	Got	
✓	100000	100000	100000	✓
✓	123	15129	15129	✓

Passou em todos os teste! ✓

Question author's solution (Python3):

```
1 def maiorQue1000(a):  
2     if a > 1000:  
3         return a  
4     else:  
5         return maiorQue1000(a**2)  
6
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 2

Correto

Atingiu 1,00 de 1,00

Soma Harmonica

Uma série harmônica é uma soma infinita, definida pela seguinte fórmula:

$$\sum_{k=1}^{\infty} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$$

Recursivamente, é possível encontrar a soma harmônica dos X primeiros elementos da série, da seguinte forma:

$$soma_harmonica(X) = \begin{cases} 1 & \text{se } X = 1 \\ \frac{1}{X} + soma_harmonica(X - 1) & \text{se } X > 1 \end{cases}$$

Dessa forma, crie uma [função](#) `soma_harmonica(X)` que calcule recursivamente a soma harmônica de $X - 1$ elementos.

Entrada

O único parâmetro da sua [função](#) será o valor $X \geq 1$.

Saída

A [função](#) deve retornar a soma harmônica de $X - 1$ elementos.

Observações

Submeta somente o que foi solicitado.

Particularidade do Tópico

Atenção, a criação de uma [função](#) com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da [função](#)).

For example:

Test	Result
<code>print(soma_harmonica(6))</code>	2.4499999999999997
<div>Aviso!</div> <code>soma_harmonica(5)</code>	2.2833333333333333

Test	Result
print(soma_harmonica(2))	1.5

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 ▼ def soma_harmonica(x):
2 ▼     if x == 1:
3         return 1
4 ▼     else:
5         return 1/x + soma_harmonica(x-1)
6

```

	Test	Expected	Got	
✓	print(soma_harmonica(6))	2.4499999999999997	2.4499999999999997	✓
✓	print(soma_harmonica(5))	2.2833333333333333	2.2833333333333333	✓
✓	print(soma_harmonica(2))	1.5	1.5	✓
✓	print(soma_harmonica(28))	3.927171038966368	3.927171038966368	✓

Aviso!

?

	Test	Expected	Got	
✓	print(soma_harmonica(74))	4.888022297219713	4.888022297219713	✓
✓	print(soma_harmonica(1))	1	1	✓

Passou em todos os teste! ✓

Question author's solution (Python3):

```
1 def soma_harmonica(n):  
2     if n < 2:  
3         return 1  
4     else:  
5         return 1 / n + (soma_harmonica(n - 1))
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 3

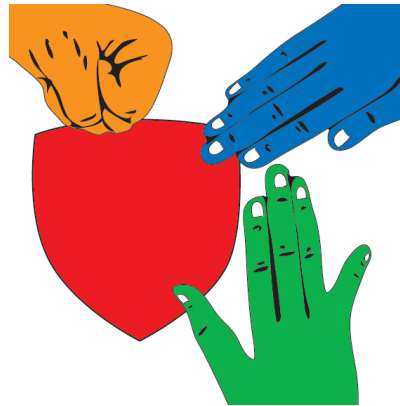
Correto

Atingiu 1,00 de 1,00

Pedra, Papel, Ataque Aéreo

Pedra, Papel, Ataque Aéreo é um jogo infantil muito popular, em que duas ou mais crianças formam um círculo e fazem gestos com a mão na tentativa de obter a vitória.

As partidas são muito simples. Os jogadores podem escolher entre o sinal de uma Pedra (o punho), o sinal de um Papel (a palma aberta), e o sinal para o Ataque Aéreo (igual o do Papel, mas com apenas o polegar e o mindinho estendidos).



Uma partida, com dois jogadores, possuem as seguintes regras para se definir um vencedor:

- Ataque Aéreo vs. Pedra: Neste caso, o jogador com o Ataque Aéreo derrota o jogador com a Pedra, por razões óbvias.
- Pedra vs. Papel: Neste caso, o jogador com a Pedra derrota o com Papel, porque a Pedra machuca muito mais.
- Papel vs. Ataque Aéreo: Aqui o Ataque Aéreo ganha, porque Ataque Aéreo sempre ganha e o Papel é patético.
- Papel vs. Papel: Nesta variação, ambos os jogadores ganham, porque o Papel é inútil e ninguém que enfrenta o Papel pode perder.
- Pedra vs. Pedra: Para este caso não há ganhador, porque depende do que os jogadores decidem fazer com a Pedra e normalmente não fazem nada.
- Ataque Aéreo vs. Ataque Aéreo: Quando isto acontece, todos os jogadores perdem, devido a Aniquilação Mútua.

Aviso!

é escrever uma função chamada $ppa(a, b)$ que, dada as escolhas de dois jogadores, informe quem venceu o jogo.

?

Entrada

A [função](#) possui dois inteiros a e b , representando as jogadas do jogador 1 e do jogador 2, que podem ser os valores 1, 2 ou 3 e estes representam o seguinte:

- 1: para representar a Pedra
- 2: para representar o Papel
- 3: para representar o Ataque Aéreo

Saída

A [função](#) deve retornar o seguinte:

- "Jogador 1 venceu": se o Jogador Um tiver vencido a partida;
- "Jogador 2 venceu": se o Jogador Dois tiver vencido a partida;
- "Ambos venceram": se os dois jogadores tiverem vencido a partida;
- "Sem ganhador": se não houver ganhador; e
- "Aniquilacao mutua": se ocorrer Aniquilação Mútua.

Observações

Submeta somente o que foi solicitado.

Particularidade do Tópico

Atenção, a criação de uma [função](#) com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da [função](#)).

For example:

Test	Result
<code>print(ppa(1, 1))</code>	Sem ganhador
<code>print(ppa(1, 2))</code>	Jogador 1 venceu
<code>print(ppa(1, 3))</code>	Jogador 2 venceu
<code>print(ppa(2, 2))</code>	Ambos venceram

Aviso!

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 ▾ def ppa(a, b):
2     # jogador 1, jogador 2
3 ▾     if a == b:
4 ▾         if a == 1:
5             return 'Sem ganhador'
6 ▾         if a == 2:
7             return 'Ambos venceram'
8 ▾         if a == 3:
9             return 'Aniquilacao mutua'
10 ▾    else:
11 ▾        if a == 3 or b == 2: #aniquilação sempre ganha e papel sempre perde
12            return 'Jogador 1 venceu'
13 ▾        elif b == 3 or a == 2:
14            return 'Jogador 2 venceu'

```

	Test	Expected	Got	
✓	print(ppa(1, 1))	Sem ganhador	Sem ganhador	✓
✓	print(ppa(1, 2))	Jogador 1 venceu	Jogador 1 venceu	✓
✓	print(ppa(1, 3))	Jogador 2 venceu	Jogador 2 venceu	✓
✓	print(ppa(2, 1))	Jogador 2 venceu	Jogador 2 venceu	✓
✓	print(ppa(2, 2))	Ambos venceram	Ambos venceram	✓
✓	print(ppa(2, 3))	Jogador 2 venceu	Jogador 2 venceu	✓

Aviso!

m todos os teste! ✓

?

Question author's solution (Python3):

```
1 def ppa(a, b):  
2     if a == 3 and b == 3:  
3         return 'Aniquilacao mutua'  
4     elif a == 2 and b == 2:  
5         return 'Ambos venceram'  
6     elif a == 1 and b == 1:  
7         return 'Sem ganhador'  
8     elif a == 3 or (a == 1 and b == 2):  
9         return 'Jogador 1 venceu'
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 4

Correto

Atingiu 1,00 de 1,00

País esquisito 2

Pedro e Guilherme passaram o ano no país esquisito e agora estão planejando continuar sua aventura! Porém para o espanto de ambos, Wil e Jon, nativos do país da água careca resolveram ir com vocês para fugir da cidade também e conhecer o mundo! Muito bom né...? Mas agora o problema se inverteu, eles não conseguem entender como o resto do mundo usa medidas tão estranhas. Como eles são muito orgulhosos, (alguns chamariam de "patriotismo") não estão dispostos a usar uma máquina conversora de medidas como Pedro e Guilherme usaram durante esse último longo ano. Eles usarão todas as medidas que encontrarem fora do país como se ela estivesse na medida de seu país. Você, como grande amigo e conterrâneo deles, fará ao menos uma [função compensa](#) que recebe dois argumentos, um inteiro n que representa o valor da medida e uma [string](#) m sendo o tipo da medida, caso o valor de n e o seu valor convertido para a medida desse país forem iguais, retorne "AHA quem precisa de conversor ?", caso contrario, retorne "Ai irmao eh melhor converter isso ai...".

- Regra 1: 1 medida de comprimento estrangeira equivale a 0.62137 medida de comprimento do país.
- Regra 2: 0° na temperatura estrangeira equivale a 32° na temperatura desse país. (use a fórmula $n * (9/5) + 32$ para fazer a conversão e use a divisão entre números reais)
- Regra 3: 5 moedas estrangeiras equivalem a 1.0 moeda deste país.

Entrada

A entrada consiste nos parâmetros da [função compensa](#), que são um número inteiro n ($0 \leq n \leq 10000$) e uma [string](#) m , o valor da medida e qual é a medida de conversão. As medidas podem ser somente 3, comprimento, temperatura e dinheiro.

Saída

Sua [função](#) deve retornar "AHA quem precisa de conversor ?" caso os valores convertidos forem iguais, ou "Ai irmao eh melhor converter isso ai...", caso contrário, ambos sem aspas.

Observações

- No primeiro caso de teste, o resultado é "AHA quem precisa de conversor ?" pois o valor convertido e o não convertido são iguais.
- No segundo caso de teste, o resultado é "Ai irmao eh melhor converter isso ai..." porque os valores convertido e não convertido são diferentes.

Aviso!

ta somente o que foi solicitado no enunciado da questão.

?

Particularidade do Tópico

Atenção, a criação de uma [função](#) com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da [função](#)).

For example:

Test	Result
<code>print(compensa(0, "comprimento"))</code>	AHA quem precisa de conversor ?
<code>print(compensa(32, "temperatura"))</code>	Ai irmao eh melhor converter isso ai...
<code>print(compensa(1000, "dinheiro"))</code>	Ai irmao eh melhor converter isso ai...

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 def compensa(n, tipo):
2     if tipo == "comprimento":
3         if n == 0:
4             return "AHA quem precisa de conversor ?"
5         else:
6             return "Ai irmao eh melhor converter isso ai..."
7     elif tipo == "temperatura":
8         if n == -40:
9             return "AHA quem precisa de conversor ?"
10        else:
11            return "Ai irmao eh melhor converter isso ai..."
12    elif tipo == "dinheiro":
13        if n == 0:
14            return "AHA quem precisa de conversor ?"
15        else:

```

Aviso!

st

Expected

Got

?

	Test	Expected	Got	
✓	<code>print(compensa(0, "comprimento"))</code>	AHA quem precisa de conversor ?	AHA quem precisa de conversor ?	✓
✓	<code>print(compensa(32, "temperatura"))</code>	Ai irmao eh melhor converter isso ai...	Ai irmao eh melhor converter isso ai...	✓
✓	<code>print(compensa(1000, "dinheiro"))</code>	Ai irmao eh melhor converter isso ai...	Ai irmao eh melhor converter isso ai...	✓
✓	<code>print(compensa(100, "temperatura"))</code>	Ai irmao eh melhor converter isso ai...	Ai irmao eh melhor converter isso ai...	✓
✓	<code>print(compensa(0, "dinheiro"))</code>	AHA quem precisa de conversor ?	AHA quem precisa de conversor ?	✓
✓	<code>print(compensa(1028, "comprimento"))</code>	Ai irmao eh melhor converter isso ai...	Ai irmao eh melhor converter isso ai...	✓

Passou em todos os teste! ✓

Para resolver o problema, deve-se converter o valor na medida correta e verificar se ambos os valores são iguais.

Question author's solution (Python3):

```

1  #Colocar aqui a solucao
2
3  def compensa(n, m):
4      if m == "comprimento":
5          aux = n*0.62137
6      elif m == "temperatura":
7          aux = n*(9/5) + 32
8      else:
9          aux = n/5
10

```

Correto

Aviso!

este envio: 1,00/1,00.

?

Questão 5

Correto

Atingiu 1,00 de 1,00

Sequência de Deivis

Dêivis está fascinado por sequências. Dessa vez, ele criou a Sequência de Dêivis que é dada pelo seguinte:

1, 2, 2, 3, 4, 6, 9, 14, 22, 35, ...

Assim, Dêivis te desafiou a criar uma [função](#) chamada **deivis_sequence(n)** que deve retornar o n -ésimo número da Sequência de Dêivis.

Entrada

O único parâmetro da sua [função](#) será o valor $1 \leq n \leq 31$, que indica a posição do número da Sequência de Deivis que você deve retornar.

Saída

A [função](#) deve retornar um inteiro: o n -ésimo número da Sequência de Dêivis.

Observações

- Submeta somente o que foi solicitado.
- Não é necessário validar se o número de entrada está dentro do intervalo definido.

Particularidade do Tópico

Atenção, a criação de uma [função](#) com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da [função](#)).

For example:

Test	Result
<code>print(deivis_sequence(2))</code>	2
<code>print(deivis_sequence(5))</code>	4
<code>print(deivis_sequence(10))</code>	35

Aviso!

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 # 1, 2, 2, 3, 4, 6, 9, 14, 22, 35, ...
2
3 def deivis_sequence(n):
4     if n == 1:
5         return 1
6     elif n == 2:
7         return 2
8     elif n == 3:
9         return 2
10    else:
11        return deivis_sequence(n-1) + fib(n-3)
12
13
14 def fib(n):
15     if n == 1 or n == 2:

```

	Test	Expected	Got	
✓	print(deivis_sequence(2))	2	2	✓
✓	print(deivis_sequence(5))	4	4	✓
✓	print(deivis_sequence(10))	35	35	✓
✓	print(deivis_sequence(15))	378	378	✓
✓	print(deivis_sequence(20))	4182	4182	✓
✓	print(deivis_sequence(25))	46369	46369	✓

Aviso!

?

Passou em todos os teste! ✓

Veja que a sequência de Devis é uma alteração da famosa sequência de Fibonnaci. Veja os termos da sequência de Fibonacci:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

e compare com os termos da sequência de Davis:

1, 2, 2, 3, 4, 6, 9, 14, 22, 35, ...

Question author's solution (Python3):

```
1 def devis_sequence(n):  
2     if n == 1:  
3         return 1  
4     if n == 2:  
5         return 2  
6
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 6

Correto

Atingiu 1,00 de 1,00

Doces

Os amigos Rofera e Loffera estão brigando e você tem que resolver. Existem n fichas de sinuca e você quer distribuí-las entre os dois amigos de um modo que:

- Loffera irá receber a ($a > 0$) fichas;
- Rofera irá receber b ($b > 0$) fichas;
- Cada amigo irá receber um número **inteiro** de fichas;
- Rofera irá receber mais fichas do que Loffera ($b > a$);
- Todas as fichas vão ser dadas para um dos dois amigos ($a + b = n$);

Sua tarefa é calcular o número de modos de distribuir exatamente n fichas entre os amigos respeitando as regras acima. Para isso, implemente a [função sinuquera\(n\)](#).

Entrada

O único parâmetro da sua [função](#) será o valor $n \geq 1$.

Saída

A [função](#) deve retornar o número de modos de distribuir exatamente n fichas entre os amigos, respeitando as regras acima.

Observações

Submeta somente o que foi solicitado no enunciado da questão.

Particularidade do Tópico

Atenção, a criação de uma [função](#) com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da [função](#)).

For example:

Test	Result
Aviso! sinuquera(7))	3

Test	Result
print(sinuquera(1))	0
print(sinuquera(456))	227

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```
1 ▾ def sinuquera(n):
2   return (n-1)//2
```

	Test	Expected	Got	
✓	print(sinuquera(7))	3	3	✓
✓	print(sinuquera(1))	0	0	✓
✓	print(sinuquera(456))	227	227	✓
Aviso!	int(sinuquera(2000000000))	999999999	999999999	✓

	Test	Expected	Got	
✓	print(sinuquera(763243547))	381621773	381621773	✓
✓	print(sinuquera(98))	48	48	✓

Passou em todos os teste! ✓

Question author's solution (Python3):

```
1 def sinuquera(n):  
2     return (n-1)//2
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 7

Correto

Atingiu 1,00 de 1,00

Uma vida musical

Gustavo e Nathan são estudantes de [Ciência da Computação](#) na UnB, mas o último semestre que eles fizeram foi muito puxado e eles estão muito cansados. Eles decidiram então, como grandes amigos, tomar um ano sabático e rodar pelo Brasil tentando fazer o que eles gostam, música. Como são estudantes, não terão muito recursos, então sempre irão trocar seus instrumentos com outros artistas para ter novas experiências e aprender novos instrumentos. Mas isso trouxe um problema, pois as vezes o novo instrumento pode não caber no carro deles. Você irá então ajudá-los! Crie um [função](#) `areaInstrumento` que irá receber três argumentos: dois inteiros n , m correspondendo ao instrumento de forma retangular usado por Nathan e um inteiro r correspondendo ao lado do triângulo equilátero que equivale ao instrumento que Gustavo toca. Calcule a área de ambos os instrumentos e retorne a soma deles.

Entrada

A entrada consiste nos parâmetros da [função](#) `areaInstrumento`, que são os três números inteiros n , m e r ($2 \leq n, m, r \leq 100$) como descritos no enunciado.

Saída

Sua [função](#) deve retornar a soma da área de ambos os instrumentos.

Observações

- No primeiro caso de teste, o resultado é `\(28.89711\)` pois a área do instrumento de Nathan é `25` e a área do instrumento do Gustavo é `3.89711`. Use o método `sqrt(x)` da [biblioteca](#) `math` para auxiliar no cálculo da área do triângulo.
- Não se preocupe com arredondamentos, o próprio juiz fará o truncamento para 5 casas decimais.
- Submeta somente o que foi solicitado no enunciado da questão.

Particularidade do Tópico

Atenção, a criação de uma [função](#) com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da [função](#)).



Test	Result
<code>print("{0:.5f}".format(areaInstrumento(5,5,3)))</code>	28.89711
<code>print("{0:.5f}".format(areaInstrumento(5,6,8)))</code>	57.71281
<code>print("{0:.5f}".format(areaInstrumento(7,20,100)))</code>	4470.12702

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 import math
2 def areaInstrumento(m, n, r):
3     return m*n + r**2*math.sqrt(3)/4

```

	Test	Expected	Got	
✓	<code>print("{0:.5f}".format(areaInstrumento(5,5,3)))</code>	28.89711	28.89711	✓
✓	<code>print("{0:.5f}".format(areaInstrumento(5,6,8)))</code>	57.71281	57.71281	✓
✓	<code>print("{0:.5f}".format(areaInstrumento(7,20,100)))</code>	4470.12702	4470.12702	✓

Aviso!

?

	Test	Expected	Got	
✓	<code>print("{0:.5f}".format(areaInstrumento(15,35,43)))</code>	1325.64049	1325.64049	✓
✓	<code>print("{0:.5f}".format(areaInstrumento(18,3,23)))</code>	283.06372	283.06372	✓
✓	<code>print("{0:.5f}".format(areaInstrumento(73,4,30)))</code>	681.71143	681.71143	✓

Passou em todos os teste! ✓

Para resolver o problema, deve-se calcular a área do retângulo e do triângulo e retornar a soma dos dois.

Question author's solution (Python3):

```
1 import math
2
3 def areaInstrumento(n, m, r):
4     areaRetangular = n*m
5     areaTriangulo = (r*r*math.sqrt(3))/4
6
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 8

Correto

Atingiu 1,00 de 1,00

Digito Zero

Dado um inteiro x escreva uma [função](#) `count_0s(x)` que retorne quantos dígitos 0 existem na representação binária de x , a partir do dígito 1 mais a esquerda do número, ou seja, o [bit](#) 1 mais significativo nessa representação.

Entrada

O único parâmetro da sua [função](#) será um inteiro $x > 0$.

Saída

A [função](#) deve retornar o número de dígitos 0 que existem na representação binária de x .

Notas

- No primeiro exemplo de teste, a representação binária da entrada $x = 6$ é 110_2 . Portanto, existe um único zero antes do [bit](#) 1 mais significativo.
- No segundo exemplo de teste, a representação binária da entrada $x = 8$ é 1000_2 . Logo, existem três [bits](#) zeros antes do único [bit](#) 1, que é o mais significativo.
- No terceiro exemplo de teste, a representação binária da entrada $x = 28$ é 10100_2 . Assim, existem três [bits](#) zeros antes do [bit](#) 1 mais significativo.

Particularidade do Tópico

Atenção, a criação de uma [função](#) com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da [função](#)).

For example:

Test	Result
Aviso! <code>count_0s(6)</code>	1

Test	Result
print(count_0s(8))	3
print(count_0s(20))	3

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 ▼ def count_0s(x):
2     zeros = 0
3 ▼     for c in str(bin(x)[2:]):
4 ▼         if c == '0':
5             zeros += 1
6     return zeros
7
8
9

```

	Test	Expected	Got	
✓	print(count_0s(6))	1	1	✓
✓	print(count_0s(8))	3	3	✓
✓	print(count_0s(20))	3	3	✓
Aviso!	print(count_0s(323))	5	5	✓

	Test	Expected	Got	
✓	print(count_0s(1024))	10	10	✓
✓	print(count_0s(1023))	0	0	✓

Passou em todos os teste! ✓

Para resolver o problema, deve-se realizar a conversão do número da entrada x para sua representação binária. Podemos fazer isso criando uma função recursiva, em que cada chamada deve gerar um bit de sua representação, bastando pegar o resto da divisão de x por 2 e contabilizar no resultado final (no comando return) a presença do bit 0. A chamada recursiva deve ter como argumento o quociente de $x \div 2$ e ter como critério de parada o caso em que o x é igual a zero.

Question author's solution (Python3):

```
1 def count_0s(x):
2     if x == 0:
3         return 1
4     elif x == 1:
5         return 0
6     else:
7         v = count_0s(x // 2)
8         if x % 2 == 0:
9             return v + 1
10        else:
11            return v
12
13 #Solucao alternativa: Prof. Vinicius Borges
14 #def count_0s(x):
15 #    if x < 1:
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 9

Correto

Atingiu 1,00 de 1,00

Pattern mais complicado

Dado um número n , crie uma função **pattern(n)** que imprima números em ordem, seguindo o padrão dos exemplos. Considere padrões de saída diferentes para números pares e ímpares.

Entrada

O único parâmetro da sua função será o valor $0 \leq n \leq 10^6$.

Saída

A função deve imprimir os valores, de acordo com o padrão dos exemplos.

Notas

- No primeiro exemplo de teste, a entrada é dada por $n = 22$ e a sequência de números na saída é 22, 17, 8, 3, 1, 0, 1, 3, 8, 17, 22. Observe que a sequência começa com o próprio 22, que é par - informação relevante para determinar o próximo número da sequência, que é 17. Em seguida, a partir do 17 que é um número ímpar, obtém-se o número 8. Os próximos números são gerados seguindo o padrão a ser determinado dependendo se cada número é par ou ímpar.

Observações

- Observe que, ao criar uma função recursiva, é possível realizar operações de impressão antes e depois da chamada recursiva, o que poderá facilitar a criação do padrão exigido pelo exercício.
- Submeta somente o que foi solicitado.

Particularidade do Tópico

Atenção, a criação de uma função com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da função).

For example:

Test	Result
Aviso!	

Test	Result
pattern(22)	22 17 8 3 1 0 1 3 8 17 22
pattern(64)	64 59 29 14 9 4 -1 4 9 14 29 59 64
pattern(7)	7 3 1 0 1 3 7

Answer: (penalty regime: 0, 0, 10, 20, ... %)

1 ▾ `def pattern(n):`

Aviso!

`print(n)`
`if n > 0:`

?

```

4 ▼   if n % 2 == 0:
5     pattern(n-5)
6     print(n)
7 ▼   else:
8     pattern(n//2)
9     print(n)

```

	Test	Expected	Got	
✓	pattern(22)	22 17 8 3 1 0 1 3 8 17 22	22 17 8 3 1 0 1 3 8 17 22	✓

Aviso!

?

	Test	Expected	Got	
✓	pattern(64)	64 59 29 14 9 4 -1 4 9 14 29 59 64	64 59 29 14 9 4 -1 4 9 14 29 59 64	✓
✓	pattern(7)	7 3 1 0 1 3 7	7 3 1 0 1 3 7	✓
✓	pattern(23)	23 11 5 2 -3 2 5 11 23	23 11 5 2 -3 2 5 11 23	✓
✓	pattern(8)	8 3 1 0 1 3 8	8 3 1 0 1 3 8	✓

Aviso!

?

	Test	Expected	Got	
✓	pattern(25)	25 12 7 3 1 0 1 3 7 12 25	25 12 7 3 1 0 1 3 7 12 25	✓

Passou em todos os teste! ✓

Veja que se um número n da sequência é par, o próximo número dessa sequência é sempre $n - 5$. Agora, se n é ímpar, repare que o próximo número da sequência é o quociente da divisão de $n \div 2$. No primeiro exemplo, $n = 22$, portanto $22 - 5 = 17$, e, por fim, $17 \div 2 = 8$, e assim, sucessivamente. Assim, podemos elaborar uma função recursiva, em que trata esses dois casos separadamente, em que as chamadas devem ocorrer apenas se $n > 0$. Assim, para imprimir a parte "crescente" da sequência, você pode imprimir o valor de n após realizar cada chamada recursiva dentro da função.

Question author's solution (Python3):

```

1 ▼ def pattern(n):
2     print(n)
3 ▼     if n > 0:
4 ▼         if n % 2 == 0:
5             pattern(n-5)
6             print(n)
7 ▼         else:
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 10

Correto

Atingiu 1,00 de 1,00

MDC

O máximo divisor comum (MDC) entre dois ou mais números inteiros é o maior número inteiro que é fator de todos eles. Por exemplo, os divisores comuns de 12 e 18 são 1, 2, 3 e 6, logo o $\text{mdc}(12, 18) = 6$. No ano 300 a.c., o livro Elementos descreve um método para encontrar o MDC de dois números que é utilizado até os dias atuais, e é conhecido como [Algoritmo](#) de Euclides. Esse método baseia-se na seguinte propriedade: $\text{mdc}(a, b) == \text{mdc}(b, r)$, onde r é o resto da divisão de a por b . Dessa forma, aproveitando-se dessa propriedade, é possível realizar sucessivas divisões entre pares de valores e seus restos, até que o resto de alguma divisão seja zero. Quando o resto passa a ser zero, o segundo elemento do último mdc realizado é o MDC dos pares iniciais. O [algoritmo](#) funciona da seguinte maneira:

$\text{mdc}(32, 26)$ - o resto da divisão de 32 por 26 é 6

$\text{mdc}(26, 6)$ - o resto da divisão de 26 por 6 é 2

$\text{mdc}(6, 2)$ - o resto da divisão de 6 por 2 é 0

$\text{mdc}(2, 0)$ - o resultado é 2, para $\text{mdc}(32, 26)$

Dessa forma, é possível escrever um [algoritmo](#) recursivo para o método de euclides da seguinte forma:

$\text{mdc}(a, b) = \begin{cases} a & \text{se } b = 0 \\ \text{mdc}(b, a \% b) & \text{se } a \% b > 0, b > 0 \end{cases}$

Dada a especificação recursiva para o [algoritmo](#) de euclides, sua tarefa é implementar uma [função](#) **mdc(a, b)** que calcule o MDC entre dois números a e b .

Entrada

A entrada consiste de dois inteiros a e b que representam os valores cujo MDC é procurado ($1 \leq a, b \leq 10^{15}$).

Saída

A [função](#) deve retornar o MDC entre os valores a e b , conforme o exemplo.

Observações

Responda somente o que foi solicitado.

Aviso!

?

- No primeiro exemplo de teste, o MDC entre 6 e 6 é calculado conforme a propriedade mencionada: $\backslash(\text{mdc}(6,6)=\text{mdc}(6,0)\backslash)$, resultando em 6.
- No segundo exemplo de teste, o MDC entre 12 e 30 é calculado conforme a propriedade mencionada: $\backslash(\text{mdc}(12,30)=\text{mdc}(30,12)=\text{mdc}(12,6)=\text{mdc}(6,0)\backslash)$, resultando em 6.
- No terceiro exemplo de teste, o MDC entre 105 e 60 é calculado conforme a propriedade mencionada: $\backslash(\text{mdc}(105,60)=\text{mdc}(60,45)=\text{mdc}(45,15)=\text{mdc}(15,0)\backslash)$, resultando em 15.

Particularidade do Tópico

Atenção, a criação de uma [função](#) com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da [função](#)).

For example:

Test	Result
<code>print(mdc(6,6))</code>	6
<code>print(mdc(12,30))</code>	6
<code>print(mdc(105,60))</code>	15

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```
1 def mdc(a, b):
2     if a > b:
3         max = int(a)
4         min = int(b)
5     else:
6         max = int(b)
7         min = int(a)
8
9     if min == 0:
10        return max
11    else:
12        return mdc(min, max % min)
```

Aviso!

?

	Test	Expected	Got	
✓	<code>print(mdc(6,6))</code>	6	6	✓
✓	<code>print(mdc(12,30))</code>	6	6	✓
✓	<code>print(mdc(105,60))</code>	15	15	✓
✓	<code>print(mdc(139,10))</code>	1	1	✓
✓	<code>print(mdc(105,460))</code>	5	5	✓
✓	<code>print(mdc(1546986256,41))</code>	1	1	✓

Passou em todos os teste! ✓

A questão pode ser resolvida elaborando-se uma função recursiva, que possui dois parâmetros a e b e que implementa o processo definido pela seguinte propriedade: $\text{mdc}(a, b) == \text{mdc}(b, r)$, onde r é o resto da divisão de a por b . Assim, se o valor de b for diferente de zero, devemos fazer uma nova chamada recursiva, denotando a operação: $\text{mdc}(a, b) == \text{mdc}(b, r)$. Agora, se b for igual a zero, encontramos o resultado em a .

Question author's solution (Python3):

```
1 def mdc(a, b):  
2     if (b == 0):  
3         return a  
4     else:
```

Correto

Aviso!

este envio: 1,00/1,00.

?



Aviso!

?