

Questão 1

Implemente uma função chamada **area** que imprime a área de uma determina figura geométrica. A função deve receber 3 parâmetros, sendo dois deles valores numéricos e uma string representando a forma geométrica. A área da figura deve ser do tipo inteiro. As formas geométricas permitidas são "retangulo" e "losango".

Entrada

Os parâmetros da função são dois inteiros $arg1, arg2 \geq 1$ e uma string *forma*. Em caso de retângulo, os argumentos representam a base e a altura da forma e caso a figura seja um losango, os argumentos representam o valor das duas diagonais.

Saída

A função deve imprimir a frase "O *forma* tem *area* de area", conforme os exemplos.

forma é a string que pode ter as formas geométricas **retangulo** ou **losango** e *area* é o valor **inteiro** do cálculo da área da forma geométrica dada na string *forma*, com *arg1*, *arg2* assumindo as incógnitas de cada cálculo de área.

For example:

Test	Result
<code>area(10, 2, 'losango')</code>	O losango tem 10 de area
<code>area(20, 4, 'retangulo')</code>	O retangulo tem 80 de area
<code>area(15, 3, 'losango')</code>	O losango tem 22 de area

Questão 2

Implemente uma função chamada **area** que imprime a área de uma determina figura geométrica. A função deve receber 3 parâmetros, sendo dois deles valores numéricos e uma string representando a forma geométrica. A área da figura deve ser do tipo inteiro. As formas geométricas permitidas são "retangulo", "losango" e "triangulo".

Entrada

Os parâmetros da função são dois inteiros $arg1, arg2 \geq 1$ e uma string *forma*. Em caso de retângulo ou triângulo, os argumentos representam a base e a altura da forma e caso a figura seja um losango, os argumentos representam o valor das duas diagonais.

Saída

A função deve imprimir a frase "O *forma* tem *area* de area".

forma é a string inserida, que pode tomar o nome de quatro formas geométricas: **retangulo**, **losango** e **triangulo** e *area* é somente o valor **inteiro** do cálculo da área da forma geométrica dada na string *forma*, com *arg1*, *arg2* assumindo as incógnitas de cada cálculo de área.

For example:

Test	Result
area(10, 2, 'losango')	0 losango tem 10 de area
area(20, 4, 'retangulo')	0 retangulo tem 80 de area
area(15, 3, 'triangulo')	0 triangulo tem 22 de area

Questão 3

Implemente uma função chamada **area** que imprime a área de uma determinada figura geométrica. A função deve receber 3 parâmetros, sendo dois deles valores numéricos e uma string representando a forma geométrica. A área da figura deve ser do tipo inteiro. As formas geométricas permitidas são retângulo, losango, triângulo e círculo.

Entrada

Os parâmetros da função são dois inteiros $arg1, arg2 \geq 1$ e uma string *forma*. Em caso de retângulo ou triângulo, os argumentos representam a base e a altura da forma, caso a figura seja um losango, os argumentos representam o valor das duas diagonais, e caso a figura geométrica seja círculo, a variável *arg1* será **sempre** o raio, e a variável *arg2* **sempre** será igual a 3.

Saída

A função deve imprimir a frase O *forma* tem *area* de area

Onde *forma* é a string inserida, que pode tomar o nome de quatro formas geométricas: **retângulo**, **losango**, **triângulo**, **círculo** e *area* é somente o valor **inteiro** do cálculo da área da forma geométrica dada na string *forma*, com *arg1*, *arg2* assumindo as incógnitas de cada cálculo de área.

Observação

Caso a figura seja um círculo, utilize *arg2* no lugar do valor de π , para calcular a área.

For example:

Test	Result
area(10, 2, 'losango')	O losango tem 10 de area
area(20, 4, 'retangulo')	O retangulo tem 80 de area
area(15, 3, 'circulo')	O circulo tem 675 de area

Questão 4

A UnBanners Eletrônicos é responsável por todos aqueles painéis eletrônicos que você encontra pelas vias de Brasília. Logo após um painel exibir uma propaganda, ele entra no modo proteção de tela, exibe alguns símbolos diversos por um tempo, para depois mostrar a próxima propaganda. O programa principal do banner já existe e funciona muito bem, porém você foi contratado para gerar quatro novos padrões de exibição para o banner. Assim, sua tarefa é implementar uma função chamada **banner** que recebe um único parâmetro n que representa um valor inteiro e exibe o padrão de acordo com a entrada exigida.

Entrada

A função **banner** recebe como parâmetro um único inteiro n ($-1000 \leq n \leq 1000$), para utilizar em cada um dos quatro tipos de padrões.

Saída

Na saída deverá ser impresso o padrão, utilizando a seguinte lógica:

- Se o número n for par e positivo, o padrão deve ser (sem aspas): "| | | | | | | |"
- Se o número n for ímpar e positivo, o padrão deve ser (sem aspas): "- - - - - - - -"
- Se o número n for par e negativo, o padrão deve ser (sem aspas): "."
- Se o número n for ímpar e negativo, o padrão deve ser (sem aspas): "= = = = = = = ="

For example:

Test	Result
banner(1)	- - - - - - - -
banner(1) banner(2)	- - - - - - - -
banner(-1) banner(1)	= = = = = = = = - - - - - - - -

Questão **5**

Em Campinas, cidade do interior de São Paulo, existem infinitos galões de água de 1 litro, que custam a dinheiros, e infinitos galões de 2 litros, que custam b dinheiros, disponíveis para venda. Qual o menor número de dinheiros necessário para Fagundes comprar **exatamente** n litros de água?

Escreva uma função chamada **dinheiros** que recebe três parâmetros referentes ao número n de Litros desejado por Fagundes, o valor a de galões de 1L e o valor b de galões de 2L. A função deve imprimir o pedido com menor valor.

Entrada

Os parâmetros da função são três inteiros $n, a, b \geq 1$.

Saída

Imprima um único inteiro com a menor quantidade de dinheiro que Fagundes precisa gastar.

For example:

Test	Result
dinheiros(10, 1, 3)	10
dinheiros(7, 3, 2)	9
dinheiros(1, 1000, 1)	1000

Questão 6

Imagine que você tenha um tabuleiro com $N \times M$ quadrados, similar a um tabuleiro de xadrez que tem 8×8 quadrados. Também imagine que você ganhou um suprimento infinito de peças de dominó, cada peça tem dimensão 2×1 e, por isso, ocupam dois quadrados do seu tabuleiro. Escreva uma função `dominos(N, M)` que imprima a quantidade máxima de dominós que podem ser colocados no tabuleiro.

Seu papel é colocar o máximo de dominós que puder no tabuleiro de modo que no fim as seguintes condições sejam satisfeitas:

- Cada dominó colocado cobre completamente dois quadrados do tabuleiro. Não pode colocar uma peça de dominó com metade dentro do tabuleiro e metade fora por exemplo.
- Nenhuma peça de dominó fica em cima de outra peça de dominó. Cada quadrado do tabuleiro só é coberta por até 1 dominó.
- Cada dominó pode ser colocado tanto na orientação vertical como na horizontal.

Entrada

Os parâmetros da função são dois inteiros $1 \leq N \leq M \leq 16$.

Saída

Um número inteiro representando a quantidade máxima de dominós que podem ser colocados no tabuleiro.

For example:

Test	Result
<code>dominos(2, 4)</code>	4
<code>dominos(3, 3)</code>	4
<code>dominos(1, 5)</code>	2

Questão **7**

Implemente a função chamada **realidade** que imprime se uma equação de segundo grau $a * x^2 + b * x + c = 0$ tem raízes reais ou complexas. A função deve receber os parâmetros a, b, e c que são números inteiros.

Entrada

Os parâmetros da função são três inteiros a, b, c .

Saída

Imprima uma única linha com a mensagem "reais", se as raízes da equação são reais; ou a mensagem "complexas", se as raízes possuem parte imaginária não nula.

For example:

Test	Result
realidade(1, 0, -1)	reais
realidade(1, 0, 1)	complexas
realidade(2, 4, 2)	reais

Questão 8

Implemente uma função chamada **formamisteriosa**, que possui como parâmetro três inteiros a, b, c , e seja capaz de identificar quais das 3 formas geométricas (quadrado, retângulo ou triângulo) podem ser formadas a partir de a e b para constituir um quadrado ou um retângulo e a, b e c para constituir um triângulo. Caso seja possível formar um triângulo, você também deve especificar se ele é escaleno, isósceles ou equilátero.

Entrada

Os parâmetros da função são três inteiros a, b, c onde $a, b, c \geq 0$.

Saída

Imprima uma linha para cada forma geométrica que a combinação desses números pode formar:

Utilizando os dois primeiros argumentos, imprima "pode ser quadrado" ou "pode ser retangulo", de acordo com a propriedade de formação da figura identificada.

Utilizando os três argumentos, imprima baseado na propriedade de triângulos e suas classificações, na mesma linha:

- pode ser triangulo – A soma de qualquer par é maior que o número restante da tripla.
- escaleno – Satisfaz a propriedade triangulo e os três lados são distintos entre si.
- isosceles – Satisfaz a propriedade triangulo e pelo menos dois lados são iguais.
- equilatero – Satisfaz a propriedade triangulo e os três lados são iguais.

For example:

Test	Result
formamisteriosa(3, 5, 4)	pode ser retangulo pode ser triangulo escaleno
formamisteriosa(2, 1, 1)	pode ser retangulo

Test	Result
formamisteriosa(3, 3, 3)	pode ser quadrado pode ser triangulo equilatero

Answer: (penalty regime: 0 %)

1 |

Questão 9

Gondim está aprendendo geometria. Ele fez uma grande lista de triplas de números e agora está tentando classificar cada tripla, determinando suas propriedades em relação a triângulos. Para terminar mais rápido, ele pediu a sua ajuda. Ele precisa que você escreva uma função chamada **classificador** que tem como parâmetro três inteiros.

Entrada

Os parâmetros da função são três inteiros a , b , c .

Saída

Imprima uma linha para cada propriedade que a tripla de números satisfaz, na ordem indicada abaixo.

- triangulo – A soma de qualquer par é maior que o número restante da tripla.
- gondim sendo gondim – Não satisfaz a propriedade triangulo.
- escaleno – Satisfaz a propriedade triangulo e os três lados são distintos entre si.
- isosceles – Satisfaz a propriedade triangulo e pelo menos dois lados são iguais.
- equilatero – Satisfaz a propriedade triangulo e os três lados são iguais.
- retangulo – Satisfaz a propriedade triangulo e o quadrado do maior lado é a soma dos quadrados dos outros lados.

Caso o triângulo possua mais de uma das propriedades estas devem ser listadas **na ordem apresentada acima**, caso o triângulo seja um triangulo isosceles e retangulo a seguinte saída é considerada errada:

```
triangulo
retangulo
isosceles
```

Enquanto a seguinte saída está correta:

```
triangulo
isosceles
retangulo
```

For example:

Test	Result
classificador(3, 5, 4)	triangulo escaleno retangulo
classificador(2, 1, 1)	gondim sendo gondim
classificador(3, 3, 3)	triangulo isosceles equilatero

Answer: (penalty regime: 0 %)

1

Questão **10**

A família Cacãope está em crise. Eles não sabem qual dos dois filhos é o mais velho! Por sorte, encontraram você, um primo distante para ajudá-los.

Escreva uma função chamada `older(ageA, ageB)` que receberá a idade em anos completos dos filhos A e B e deverá imprimir A se o filho A for com certeza mais velho, B se o filho B for com certeza mais velho e `Maybe twins` se não for possível saber com certeza é o mais velho.

Entrada

A entrada consiste de dois inteiros *ageA*, *ageB* que indicam a idade em anos completos dos filhos A e B, respectivamente.

Saída

A saída deve conter A se o filho A for com certeza mais velho, B se o filho B for com certeza mais velho e `Maybe twins` se não for possível saber com certeza é o mais velho.

For example:

Test	Result
<code>older(5, 10)</code>	B
<code>older(19, 17)</code>	A
<code>older(44, 44)</code>	Maybe twins

Questão **11**

O MMORPG mais famoso do momento, Instant Soul Crushing (ISC), permite que seus jogadores realizem duelos entre si sob o sistema RISCU. Este sistema recebe o nível de poder dos dois jogadores e diz qual jogador é o ganhador e qual jogador é o perdedor.

Você, como um membro de ISC, deve implementar o sistema RISCU: escreva uma função chamada `riscu(powerA, powerB)` que receberá o poder dos jogadores A e B e deverá imprimir:

- Se o jogador A for mais forte: Jogador A vence
- Se o jogador B for mais forte: Jogador B vence
- Se ambos jogadores têm a mesma força: Dois jogadores igualmente fracos

Entrada

A entrada consiste de dois inteiros $powerA$, $powerB$ que indicam os poderes dos jogadores A e B, respectivamente.

Saída

A saída deve conter a frase Jogador A vence se o jogador A for mais forte, Jogador B vence se o jogador B for mais forte ou Dois jogadores igualmente fracos se ambos jogadores possuem a mesma força.

For example:

Test	Result
<code>riscu(5, 10)</code>	Jogador B vence
<code>riscu(7231, 4442)</code>	Jogador A vence
<code>riscu(457278, 457278)</code>	Dois jogadores igualmente fracos

Questão **12**

Você foi convidado para jogar um jogo. Nesse jogo, você irá receber dois inteiros a e b . Em uma jogada, você pode escolher qualquer **inteiro** k entre 1 até 10 e adicionar em a ou subtrair de a . Em outras palavras, você escolhe um inteiro $k \in [1; 10]$ e executa $a := a + k$ ou $a := a - k$. Você pode usar valores diferentes de k em diferentes jogadas se quiser.

Seu papel é encontrar o número **mínimo** de jogadas para transformar a em b .

Escreva uma função **jogadas(a, b)** que imprima o que lhe é pedido.

Entrada

Os parâmetros da função são dois inteiros $a, b \geq 1$.

Saída

Imprima um número - o número mínimo de jogadas para transformar a em b .

For example:

Test	Result
jogadas(5, 5)	0
jogadas(13, 42)	3
jogadas(18, 4)	2

Questão **13**

Na maior seca da história de Brasília, Lamá e Ladê decidiram comprar uma melancia. Eles escolheram a maior e mais madura que tinha, na opinião deles. Eles pesaram a melancia, viram que ela pesava w kg e então correram para casa, morrendo de sede. Ao chegar, foram dividir a melancia mas encontraram um grande problema.

Lamá e Ladê, fãs de matemática desde cedo, amam números pares. Por isso, eles querem dividir a melancia de um modo que cada uma das duas partes pese um exato número par de kilogramas. Eles estão extremamente cansados e querem comer o quanto antes, logo eles decidiram pedir a ajuda do universitário, você, para descobrir se podem dividir a melancia do jeito que eles querem.

Claro, o peso de cada pedaço tem que ser um número inteiro maior do que 0.

Escreva a função **sedeDeMelancia(w)**, para resolver o problema requisitado.

Entrada

A entrada consiste de um inteiro $1 \leq w \leq 100$ que indica o peso da melancia trazida pelos dois.

Saída

Imprima "SIM" se os dois podem dividir a melancia em duas partes, cada uma delas pesando um número par de kg. Imprima "NAO" caso não seja possível.

For example:

Test	Result
sedeDeMelancia(8)	SIM
sedeDeMelancia(15)	NAO
sedeDeMelancia(24)	SIM

Questão **14**

Antes da pandemia, durante uma das maiores secas da história de Brasília, quatro calourinhos decidiram contratar Dêvis para embarcar em uma jornada épica em busca de uma iguaria divina, conhecida apenas por lendas urbanas como O Açaí do Ceubinho.

Como todos os calourinhos ainda não são muito amigos, e portanto não possuem tanta intimidade, eles pediram para Dêvis comprar uma quantidade de copos que fosse divisível por 4, para que cada um pudesse comer o seu sem ter que dividir a refrescância com o coleguinha. O problema é que Dêvis não é muito bom com matemática, ou com realizar tarefas corretamente, então antes de levar O Açaí do Ceubinho, ele decidiu levar pra você decidir se a quantidade de copos de açaí está correta!

Escreva a função **qtdcopos(n)** que faça o pedido.

Entrada

A entrada consiste de um inteiro $0 \leq n \leq 100$ que indica quantos copos ele levou para você avaliar.

Saída

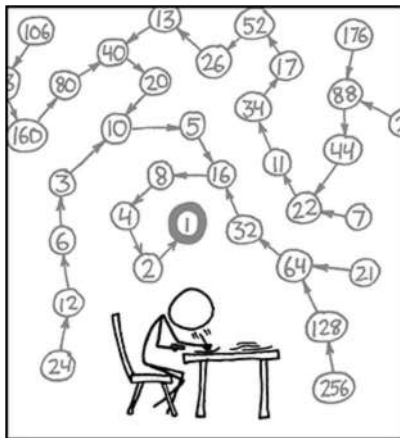
Imprima "Pode levar pros calourinhos, deivis!" se a quantidade de copos for divisível por 4. Imprima "Pode voltar pro ceubinho, deivis! Falta(m) x copo(s)!" onde x é o número de copos restantes para que o número se torne divisível por 4, caso não seja possível dar copos para todos os calourinhos.

For example:

Test	Result
qtdcopos(8)	Pode levar pros calourinhos, deivis!
qtdcopos(15)	Pode voltar pro ceubinho, deivis! Falta(m) 1 copo(s)!
qtdcopos(0)	Pode voltar pro ceubinho, deivis! Falta(m) 4 copo(s)!

Questão 15

Números Maravilhosos constituem um problema muito bem conhecido no meio da matemática. Esse problema tem diversos outros nomes e um deles é a Conjectura de Siracusa. Uma conjectura em matemática é uma proposição ou sentença tida como verdade que, no entanto, nunca foi provada. Quando é provada ser verdadeira ela passa a ser um Teorema. A Conjectura de Siracusa tem o seguinte enunciado: **"A partir de qualquer número n , dividindo-o por 2 se for par, ou multiplicando por 3 e adicionando 1 se for ímpar, e fazendo assim sucessivamente, chegaremos sempre ao número 1"**.



De uma maneira recursiva, esse problema pode ser visto da seguinte forma:

$$maravilhosos(x) = \begin{cases} 1 & \text{se } x = 1 \\ maravilhosos(3x + 1) & \text{se } x \text{ for impar} \\ maravilhosos(\frac{x}{2}) & \text{se } x \text{ for par} \end{cases}$$

Dessa forma, implemente um programa para receber um valor x e imprimir a sequência de números que x irá tomar até que seu valor alcance 1. Seu programa deve possuir uma função recursiva chamada **maravilhosos** com um parâmetro chamado **x** que irá receber um valor e deverá imprimir recursivamente a aplicação da função até que o caso base (valor 1) seja alcançado.

Entrada

de 1 inteiro x tal que $(1 < x < 10^5)$.

Cada linha da saída deve possuir um inteiro da sequência até o número um, conforme os exemplos.

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, por esse critério, caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da função).

For example:

Input	Result
4	4 2 1
5	5 16 8 4 2 1
7	7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Questão 16

Agora que o Prof. Nerynho já construiu sua piscina, ele está testando um programa de localização pessoal que diz o que Nerynho está fazendo na área de sua piscina, baseado em uma representação cartesiana vista de um satélite!!!! Ajude o Prof Nerynho a desenvolver seu programa, escrevendo a função **piscininha(x, y, w, h, a, b)** cujas variáveis são:

- As primeiras quatro variáveis, $x, y, w, h \mid w, h \geq 2$ representam a piscina, que é um retângulo de altura h e largura w , alinhado aos eixos cartesianos **X** e **Y**, cujo vértice inferior esquerdo está no ponto (x, y) .
- As duas ultimas variáveis, a e b , representam as coordenadas cartesianas (a, b) de onde o Prof Nerynho está.

Entrada

Os parâmetros da função são seis inteiros x, y, w, h, a, b .

Saída

A saída depende da posição do Prof. Nerynho baseado em onde está sua piscina e:

- Caso o professor esteja dentro da piscina, o programa deverá imprimir a frase **Dando um tchibum**;
- Caso o professor esteja fora da piscina, o programa deverá imprimir a frase **Tomando um solzin**;
- Caso o professor esteja na borda da piscina, o programa deverá imprimir a frase **So com os pezin dentro da agua**.

For example:

Test	Result
piscininha(0, 0, 10, 10, 0, 5)	So com os pezin dentro da agua
piscininha(0, 0, 10, 10, 1, 1)	Dando um tchibum
piscininha(-10, -15, 5, 4, 0, 0)	Tomando um solzin

Questão 17

O buffet de Seu Pimenta é muito famoso na cidade onde mora e, por isso, seus serviços são muito solicitados nos eventos locais. Nos últimos dias, devido à realização de vários eventos em um mesmo dia, Seu Pimenta encontrou dificuldades para orientar adequadamente seus funcionários. Como resultado, Seu Pimenta têm notado que alguns talheres "sumiram" ou que não foram corretamente organizados e armazenados, deixando o depósito e o armário de talheres bastante bagunçado.

Recentemente, Seu Pimenta recebeu uma proposta para preparar um banquete para uma conferência famosa, que receberá muitas pessoas de fora da cidade. Considerando os últimos fatos relacionados aos talheres do buffet, Seu Pimenta deixou claro ao cliente que não tem talheres suficientes para atender todos os participantes. No entanto, confiando em Seu Pimenta e na sua alta credibilidade na cidade, o cliente solicitou seus serviços mesmo assim e Seu Pimenta agora tem que correr para organizar mais um buffet.

Seu Pimenta consultou o depósito e verificou que existiam números diferentes de garfos, facas e colheres. Para maximizar a quantidade de pessoas que conseguirão jantar, Seu Pimenta estabeleceu que cada pessoa seria capaz de jantar se possuísse um par de garfo e faca, ou uma colher. Ajude Seu Pimenta e elabore uma função chamada **quantosJantam(n , g , f , c)** que calcule a maior quantidade possível de pessoas presentes no banquete que conseguiriam jantar de acordo com a estratégia proposta.

Entrada

Os parâmetros da função são quatro inteiros $n, g, f, c \geq 0$ indicando, respectivamente, a quantidade de pessoas existentes no banquete, a quantidade de garfos, a quantidade de facas e a quantidade de colheres.

Saída

Imprima um valor inteiro indicando a maior quantidade possível de pessoas presentes no banquete que conseguirão jantar.

For example:

Test	Result
quantosJantam(10, 3, 1, 4)	5
quantosJantam(15, 2, 5, 2)	4
quantosJantam(12, 0, 1, 7)	7

Questão **18**

Um triângulo é dito Pitorestico se o tamanho de todos os seus três lados são inteiros e divisíveis por 2, 3 e 5. Dados os tamanhos dos três lados de um triângulo, implemente uma função chamada **pitorestico** que recebe os três lados do triângulo **a**, **b**, **c** e imprima o que lhe é pedido.

Entrada

Os parâmetros da função são três inteiros $a, b, c \geq 1$.

Saída

Imprima "Pitorestico!!!" caso o triângulo formado pelos lados a, b, c seja pitorestico e "Nao foi dessa vez" caso contrário.

For example:

Test	Result
<code>pitorestico(30, 60, 60)</code>	Pitorestico!!!
<code>pitorestico(30, 40, 60)</code>	Nao foi dessa vez
<code>pitorestico(1, 1, 1)</code>	Nao foi dessa vez