

### **Questão 01 (Valor 2,00 ponto) (OBI2009)**

Para descontrair os alunos após as provas da OBI, a Diretora da escola organizou um campeonato de aviões de papel. Cada aluno participante receberá uma certa quantidade de folhas de um papel especial para fazer os seus modelos de aviões. A quantidade de folhas que cada aluno deverá receber ainda não foi determinada: ela será decidida pelos juízes do campeonato.

A diretora convidou, para atuarem como juízes, engenheiros da Embraer, uma das mais bem sucedidas empresas brasileiras, que vende aviões com tecnologia brasileira no mundo todo. O campeonato está programado para começar logo após a prova da OBI, mas os juízes ainda não chegaram à escola. A diretora está aflita, pois comprou uma boa quantidade de folhas de papel especial, mas não sabe se a quantidade comprada vai ser suficiente.

Considere, por exemplo, que a Diretora comprou 100 folhas de papel especial, e que há 33 competidores. Se os juízes decidirem que cada competidor tem direito a três folhas de papel, a quantidade comprada pela diretora é suficiente. Mas se os juízes decidirem que cada competidor tem direito a quatro folhas, a quantidade comprada pela diretora não seria suficiente.

Você deve escrever uma função chamada **avioes\_de\_papel** que, dados o número de competidores, o número de folhas de papel especial compradas pela Diretora e o número de folhas que cada competidor deve receber, determine se o número de folhas comprado pela Diretora é suficiente.

A sua função vai receber três números inteiros como argumentos. O primeiro argumento representa o número de competidores, o segundo argumento a quantidade de folhas de papel especial compradas pela Diretora e o terceiro argumento a quantidade de folhas de papel especial que cada competidor deve receber.

Sua função deve retornar *True* se a quantidade de folhas compradas pela Diretora é suficiente, ou *False* caso contrário.

Exemplos:

Chamada da função	Retorno da função
avioes_de_papel (10,100,10)	True
avioes_de_papel (10,90,10)	False
avioes_de_papel (5,40,2)	True

### Questão 02 (Valor 2,00 ponto) (OBI2009)

Ido é um garoto muito esperto que adora promoções e sorteios. Como já participou de muitas promoções da forma *`para participar, envie n rótulos de produtos ...'*, Aldo tem o costume de guardar o rótulo de todos os produtos que compra. Dessa forma, sempre que uma empresa faz uma promoção ele já tem um monte de rótulos para mandar.

A SBC (Super Balas e Caramelos) está fazendo uma nova promoção, e, como era de se esperar, Aldo quer participar. Para participar da promoção é preciso enviar um envelope contendo um rótulo de cada tipo de bala que a SBC produz. Por exemplo, se a SBC produz 3 tipos de balas, A, B, C, e uma pessoa tem 3 rótulos de A, 3 de B e 2 de C, ela pode enviar no máximo 2 envelopes, já que falta um rótulo de C para compor o terceiro envelope. Não há limite para o número de envelopes que uma pessoa pode enviar.

Balas são a segunda coisa de que Aldo mais gosta (a primeira como você sabe são promoções). Por causa disso a quantidade de rótulos de balas que ele tem é muito grande, e ele não está conseguindo determinar a quantidade máxima de envelopes que ele pode enviar.

Como você é o melhor amigo de Aldo ele pediu sua ajuda para fazer o cálculo, de modo que ele compre o número exato de envelopes. Você deve escrever uma função chamada **numero\_envelopes** que, a partir da lista de rótulos de Aldo, calcula o número máximo de envelopes válidos que ele pode enviar.

A sua função vai receber uma lista de números inteiros como o único argumento dela. Cada número nessa lista representa uma quantidade de rótulos de balas que Aldo tem. O primeiro número nessa lista representa a quantidade de rótulos do tipo 1 que Aldo possui, o segundo número representa a quantidade de rótulos do tipo 2, e assim por diante, até o último número.

Sua função deve retornar um número inteiro que represente o número máximo de envelopes válidos que Ido pode enviar.

Exemplos:

Chamada da função	Retorno da função
<code>numero_envelopes([5,3,6,2])</code>	2
<code>numero_envelopes([10,5,21,3,0,11])</code>	0

### Questão 03 (Valor 2,00 ponto)(OBI2009)

Os computadores foram inventados para realizar cálculos muito rapidamente, e atendem a esse requisito de maneira extraordinária. Porém, nem toda conta pode ser feita num computador, pois ele não consegue representar todos os números dentro de sua memória. Em um computador pessoal atual, por exemplo, o maior inteiro que é possível representar em sua memória é 4.294.967.295. Caso alguma conta executada pelo computador dê um resultado acima desse número, ocorrerá o que chamamos de *overflow*, que é quando o computador faz uma conta e o resultado não pode ser representado, por ser maior do que o valor máximo permitido (em inglês *overflow* significa transbordar). Por exemplo, se um computador só pode representar números menores do que 1023 e mandamos ele executar a conta  $1022 + 5$ , vai ocorrer *overflow*.

Você deve escrever uma função chamada **overflow** que recebe dois argumentos. O primeiro argumento é um número inteiro representando o maior número que um computador consegue representar. O segundo argumento é uma string que é construída da seguinte maneira:

- Começa com um inteiro P, seguido de um espaço em branco, seguido de um caractere C (que pode ser "+" ou "\*", representando os operadores de adição e multiplicação, respectivamente), seguido de um espaço em branco, seguido de um outro inteiro Q.
- A string representa a expressão  $P + Q$ , se o caractere C for "+", ou  $P \times Q$ , se o caractere C for "\*".

A sua função deve retornar True se o resultado da expressão causar um *overflow*, ou False caso contrário.

Exemplos:

Chamada da função	Retorno da função
<code>overflow(10,'5 + 5')</code>	True
<code>overflow(44,'23 * 2')</code>	False
<code>overflow(323500,'42 * 35')</code>	True

#### Questão 04 (Valor 2,00 ponto)(OBI2010)

Parte do treinamento de um novo garçom é carregar uma grande bandeja com várias latas de bebidas e copos e entregá-las todas numa mesa do restaurante. Durante o treinamento é comum que os garçons deixem cair as bandejas, quebrando todos os copos.

A SBC -- Sociedade Brasileira de Copos -- analisou estatísticas do treinamento de diversos garçons e descobriu que os garçons em treinamento deixam cair apenas bandejas que têm mais latas de bebidas que copos. Por exemplo, se uma bandeja tiver 10 latas e 4 copos, certamente o garçom em treinamento a deixará cair, quebrando os 4 copos. Já se a bandeja tiver 5 latas e 6 copos, ele conseguirá entregá-la sem deixar cair.

Você deve escrever uma função chamada **garcom** que, dado o número de latas e copos em cada bandeja que o garçom tentou entregar, imprime o total de copos que ele quebrou.

A sua função vai receber dois argumentos. O primeiro argumento é um inteiro N representando o número de bandejas que o garçom tentou entregar. O segundo argumento é uma lista de tamanho N aonde os elementos são tuplas representando as bandejas que o garçom carregou, cada tupla contém dois inteiros L e C, indicando o número de latas e o número de copos naquela bandeja, respectivamente.

Sua função deve retornar um único inteiro, indicando o número total de copos que o garçom quebrou.

Exemplos:

Chamada da função	Retorno da função
garcom (3, [(10,5),(6,8),(3,3)])	5
garcom (4, [(10,6),(8,8),(5,1),(100,100)])	7

### Questão 05 (Valor 2,00 ponto)(OBI2010)

A Copa do Mundo de 2010 será realizada na África do Sul. Bolas de futebol são muito fáceis de transportar, já que elas saem das fábricas vazias e só são enchidas somente pelas lojas ou pelos consumidores finais. Infelizmente o mesmo não pode ser dito das bolas de boliche. Como elas são completamente sólidas, elas só podem ser transportadas embaladas uma a uma, em caixas separadas.

A SBC -- Só Boliche Cascavel -- é uma fábrica de bolas de boliche que trabalha somente através de encomendas e envia todas as bolas por SEDEX. Como as bolas têm tamanhos diferentes, a SBC tem vários tamanhos de caixas diferentes para transportá-las.

Escreva uma função chamada **sedex** que, dado o diâmetro de uma bola e as 3 dimensões de uma caixa (altura, largura e profundidade), diz se a bola de boliche cabe dentro da caixa ou não.

A sua função vai receber quatro números inteiros como argumentos. O primeiro argumento é um inteiro que indica o diâmetro da bola de boliche. O segundo argumento é um inteiro que representa a altura da caixa. O terceiro argumento é um inteiro que representa a largura da caixa. O quarto argumento é um inteiro que representa a profundidade da caixa.

Sua função deve retornar True caso a bola de boliche caiba dentro da caixa ou False caso contrário.

Exemplos:

Chamada da função	Retorno da função
sedex(3,2,3,5)	False
sedex(5,5,5,5)	True
sedex(9,15,9,10)	True

**Questão 06 (Valor 2,00 ponto)(OBI2018)**

Dizemos que uma sequência de números é uma *escadinha*, se a diferença entre números consecutivos é sempre a mesma. Por exemplo, "2, 3, 4, 5" e "10, 7, 4" são escadinhas. Note que qualquer sequência com apenas um ou dois números também é uma escadinha!

Neste problema estamos procurando escadinhas em uma sequência maior de números. Dada uma sequência de números, queremos determinar quantas escadinhas existem. Mas só estamos interessados em escadinhas tão longas quanto possível. Por isso, se uma escadinha é um pedaço de outra, consideramos somente a maior. Por exemplo, na sequência "1, 1, 1, 3, 5, 4, 8, 12" temos 4 escadinhas diferentes: "1, 1, 1", "1, 3, 5", "5, 4" e "4, 8, 12".

Escreva uma função chamada **escadinha** que recebe dois argumentos. O primeiro argumento é um número inteiro que representa o tamanho da lista da sequência que queremos verificar. O segundo argumento é uma lista de números inteiros que é a sequência aonde queremos procurar as escadinhas.

Como resultado a sua função deve retornar um número inteiro representando quantas escadinhas existem na sequência.

Exemplos:

Chamada da função	Retorno da função
<code>escadinha(8,[1,1,1,3,5,4,8,12])</code>	4
<code>escadinha(1,[112])</code>	1
<code>escadinha(5,[11,-106,-223,-340,-457])</code>	1