

Questão 01 (Valor 2,00 ponto) (OBI2008)

O principal prêmio da Olimpíada Brasileira de Informática é o convite para os cursos de programação oferecidos no Instituto de Computação da Unicamp, com todas as despesas pagas pela Fundação Carlos Chagas, patrocinadora da OBI. São convidados apenas os competidores que atingem um certo número mínimo de pontos, consideradas as duas fases de provas. Você foi contratado pela Coordenação da OBI para fazer um programa que, dados os números de pontos obtidos por cada competidor em cada uma das fases, e o número mínimo de pontos para ser convidado, determine quantos competidores serão convidados para o curso na Unicamp. Você deve considerar que

- todos os competidores participaram das duas fases;
- o total de pontos de um competidor é a soma dos pontos obtidos nas duas fases.

Por exemplo, se a pontuação mínima para ser convidado é 435 pontos, um competidor que tenha obtido 200 pontos na primeira fase e 235 pontos na segunda fase será convidado para o curso na Unicamp. Já um competidor que tenha obtido 200 pontos na primeira fase e 234 pontos na segunda fase não será convidado.

Você deve escrever uma função chamada **qtd_competidores** que recebe três argumentos. O primeiro argumento é um número inteiro N representando o número de competidores, o segundo é um número inteiro P representando o número mínimo de pontos para ser convidado, o terceiro argumento é uma lista de tamanho N, onde cada elemento da lista é uma tupla de dois números X e Y indicando a pontuação de um competidor em cada uma das fases.

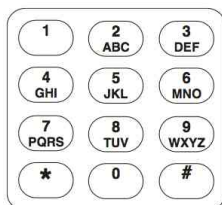
Sua função deve retornar um único número inteiro indicando o número de competidores que serão convidados a participar do curso na Unicamp.

Exemplos:

Chamada da função	Retorno da função
<code>qtd_competidores (3, 100, [(50,50),(100,0),(49,50)])</code>	2
<code>qtd_competidores (4, 235, [(100,134),(0,0),(200,200),(150,150)])</code>	2

Questão 02 (Valor 2,00 ponto) (OBI2008)

As primeiras redes públicas de telefonia foram construídas pela AT&T; no começo do século XX. Elas permitiam que seus assinantes conversassem com a ajuda de uma telefonista, que conectava as linhas dos assinantes com um cabo especial. Essas redes evoluíram muito desde então, com a ajuda de vários avanços tecnológicos. Hoje em dia, essas redes atendem centenas de milhões de assinantes; ao invés de falar diretamente com uma telefonista, você pode simplesmente discar o número da pessoa desejada no telefone. Cada assinante recebe um número de telefone -- por exemplo, 55-98-234-5678. Qualquer pessoa que discar esse número consegue então falar com a pessoa do outro lado da linha. Os hífen no número de telefone são só para facilitar a leitura, e não são discados no telefone. Para que fique mais fácil de se lembrar de um número de telefone, muitas companhias divulgam números que contém letras no lugar de dígitos. Para convertê-los de volta para dígitos, a maioria dos telefones tem letras nas suas teclas:



Ao invés de discar uma letra, discar-se a tecla que contém aquela letra. Por exemplo, se você quiser discar o número 0800-FALE-SBC, você na realidade discaria 0800-3253-722. A sua avó tem reclamado de problemas de vista -- em particular, ela não consegue mais enxergar as letrinhas nas teclas do telefone, e por isso queria que você fizesse um programa que convertesse as letras em um número de telefone para dígitos.

Você deve escrever uma função chamada **telefone** que recebe como argumento apenas uma string contendo o número de telefone que deve ser traduzido. O número de telefone contém entre 1 e 15 caracteres, que podem ser letras de A a Y e hífen (-).

Seu programa deve retornar uma string com o número de telefone com as letras convertidas para dígitos. Hífen no número de telefone devem ser mantidos no número de telefone de saída.

Exemplos:

Chamada da função	Retorno da função
telefone('PIPOCA')	'747622'
telefone('FALE-SBC')	'3253-722'

Questão 03 (Valor 2,00 ponto)(OBI2008)

A maioria das universidades brasileiras usa o vestibular para selecionar seus alunos. O vestibular consiste de uma ou mais provas sobre as matérias do Ensino Médio, visando avaliar os conhecimentos dos candidatos. Um formato popular de prova de vestibular é a prova objetiva. Neste formato de prova, cada candidato deve escolher uma das cinco alternativas apresentadas pela questão como sendo a correta. Durante a correção dos cartões, cada questão onde a alternativa escolhida pelo candidato é a mesma do gabarito, ele ganha um ponto. Alguns dos vestibulares mais concorridos do Brasil são disputados por dezenas de milhares de candidatos, e, por isso, geralmente usa-se uma folha de leitura óptica e um programa de computador para corrigir as provas de todos os candidatos e gerar a lista com suas pontuações. Você trabalha no comitê responsável pelo vestibular em uma faculdade e deve escrever um programa que, dado o gabarito e as respostas de um dos candidatos, determina o número de acertos daquele candidato.

Você deve escrever uma função chamada **vestibular** que recebe três argumentos. O primeiro argumento é um inteiro N, indicando o número de questões da prova. O segundo argumento é uma string de tamanho N indicando o gabarito da prova. O terceiro argumento é outra string de tamanho N indicando as opções marcadas pelo candidato. Ambas as strings contêm apenas os caracteres 'A', 'B', 'C', 'D' e 'E' (sempre em letra maiúscula).

Sua função deve retornar na saída padrão um número inteiro, indicando o número de acertos do candidato.

Exemplos:

Chamada da função	Retorno da função
vestibular(7,'AEDBCCE','ADDCCBE')	4
vestibular(5,'ABCDE','ABCDE')	5
vestibular(10,'ABCDEABCDE','BCDEABCDEA')	0

Questão 04 (Valor 3,00 ponto)(OBI2008)

Certas regiões resolveram o problema de tráfego intenso com a construção de auto estradas, que são estradas contendo em geral quatro ou mais pistas de rolagem em cada sentido, de forma que um número grande de carros possa passar sem que ocorram congestionamentos. O problema das auto estradas é que, junto com os carros temos um aumento considerável de ruído nas imediações da pista, o que incomoda os moradores das regiões próximas.

A GoTo engenharia, uma empresa do ramo de construção especializada em obras de estradas, encontrou uma solução engenhosa para o problema: instalar grandes painéis defletores de som de cada lado da auto estrada para tentar minimizar o ruído percebido pelos vizinhos.

Os painéis são construídos em blocos contínuos de 10 metros lineares. A auto estrada também é dividida em blocos de 10 metros de extensão, sendo cada bloco descrito por um código, como definido abaixo:

- **P** - Pista, trecho em linha reta sem curvas ou saídas. Deve-se instalar um painel de cada lado da auto estrada.
- **C** - Curva, trecho em curva de 90 graus na auto estrada. Deve-se instalar dois painéis de concreto do lado externo da curva; o outro lado fica sem painel instalado.
- **A** - Acesso, trecho em linha reta no qual existe uma entrada ou uma saída a partir de um dos lados da auto estrada (mas não do outro). Deve-se instalar um painel no lado onde **não** existe o acesso.
- **D** - Duplo acesso, trecho em linha reta no qual existem dois acessos (entradas ou saídas, em qualquer combinação possível), um de cada lado da rodovia. Nenhum painel deve ser instalado nesse bloco da auto estrada.

Apesar de ser uma empresa formada por engenheiros, nenhum dos funcionários da GoTo sabe programar, de forma que eles decidiram contrataram você como consultor independente. Você deve escrever um programa para, dado um mapa da auto estrada, determinar quantos painéis defletores são necessários para cobrir toda a extensão dessa auto estrada.

Você deve fazer uma função chamada **auto_estrada** que recebe dois argumentos. O primeiro argumento é um número inteiro C, indicando o comprimento da auto estrada, em blocos de 10 metros. O segundo argumento é uma string de tamanho C, cada letra da string descrevendo um bloco de 10 metros da auto estrada, como definido acima.

Sua função deve retornar um número inteiro, representando quantas unidades de painel são necessárias para cobrir toda a extensão da auto estrada.

Exemplos:

Chamada da função	Retorno da função
auto_estrada (5, 'DAPCD')	5
auto_estrada (8, 'AACCAAPP')	12
auto_estrada (14, 'ADCCPPPPPAADCP')	21

Questão 05 (Valor 3,00 ponto)(OBI2008)

Juninho é um menino muito visionário e inteligente, que quer ficar rico. Aos 12 anos de idade, já está interessado em investimentos de ações na bolsa de valores. Uma ação é como se fosse um pedaço de uma empresa que qualquer pessoa pode comprar. E como cada empresa tem tamanhos e valores diferentes, cada ação também tem diferentes valores, e esses valores mudam o tempo todo. Suponha, por exemplo, que Juninho compre uma ação de uma empresa de petróleo, que custe R\$ 100. Suponha que no dia seguinte, essa empresa descubra um enorme poço de petróleo, o que vai dar muitos lucros para ela no futuro. Essa empresa, então, passa a ser mais valorizada, e consequentemente o preço das ações sobem. Suponha que as ações subiram 20% nesse dia. Então Juninho, que tinha uma ação de R\$ 100, hoje tem a mesma ação, mas que vale R\$ 120. Ou seja, se ele a vender hoje, vai ter um lucro de R\$ 20, só por ter comprado e vendido a ação. Uma empresa de refrigerantes criou um novo tipo de investimento especial para iniciantes. Ela ocorre da seguinte maneira:

- O investidor compra as ações da empresa na manhã do dia X;
- O dinheiro fica investido durante exatamente quatro dias seguidos;
- Ao final dos quatro dias, são aplicados juros simples ao preço das ações; todas elas são vendidas e o dinheiro é dado de volta ao investidor.

Por exemplo, suponha que as variações do preço das ações sejam:

- Dia 1: Aumento de 3%
- Dia 2: Aumento de 1%
- Dia 3: Queda de 2%
- Dia 4: Queda de 3%
- Dia 5: Aumento de 5%
- Dia 6: Queda de 5%

Se aplicarmos R\$ 100 no dia 1, ao final do dia 4 vamos ter uma variação de $3+1-2-3 = -1$, ou seja, prejuízo de R\$ 1. Mas se começarmos aplicando no dia 2, ao final teremos uma variação de $1-2-3+5 = 1\%$, ou seja, lucro de R\$ 1. Juninho, que além de inteligente é também vidente (ou seja, consegue prever o futuro), pediu a sua ajuda para descobrir qual é a maior quantidade de dinheiro que ele pode lucrar investindo exatamente R\$ 100,00 durante quatro dias. Para isso, ele vai te dizer a variação das ações nos próximos N dias seguidos, onde $N \geq 4$.

Você deve escrever uma função chamada **acoes_bolsa** que recebe como único argumento uma lista de números inteiros indicando a variação do preço das ações, dia a dia.

Sua função deve retornar um número inteiro indicando qual é o maior lucro que Juninho pode conseguir ao investir nos dias que ele previu (o dinheiro não pode ficar investido em nenhum dia que ele não previu). Note que o "lucro" pode na verdade ser prejuízo (lucro negativo), se as ações se desvalorizarem.

Exemplos:

Chamada da função	Retorno da função
<code>acoes_bolsa([3,1,-2,-3,5])</code>	1
<code>acoes_bolsa([-2,1,-2,1,0,-3])</code>	0
<code>acoes_bolsa([-1,-2,-3,-4,-5,-6,-7,-8])</code>	-10

Questão 06 (Valor 3,00 ponto)(OBI2008)

1. Arthur é um menino pobre, e por isso tudo o que ele tem é de qualidade inferior. Porém, Arthur é uma pessoa muito inteligente e dedicada, e portanto está na escola, para poder ter uma boa educação e conseguir mudar essa situação. Atualmente, Arthur está estudando divisão, na matéria de matemática. Na hora fazer exercícios, os alunos fazem uso de uma calculadora para verificar se o que fizeram está correto. Como sabemos, Arthur não tem muito dinheiro, logo a calculadora que ele tem não é muito boa. Tal calculadora reconhece apenas números com N bits, ou seja, ela não é capaz de fazer contas com números maiores ou iguais a 2^N . Arthur, por ser inteligente, consegue, na maioria das vezes, contornar esse problema de uma maneira muito perspicaz. Por exemplo, suponha que arthur precise fazer o cálculo da divisão $200/90$ (duzentos dividido por noventa), mas que sua calculadora seja de apenas 5 bits (isto é, não consegue trabalhar com números maiores ou iguais que $2^5 = 32$). Ele sabe que se dividir o dividendo e o divisor por 10, o resultado continuará o mesmo. Então, ele faz o cálculo de $20/9$, e consegue o resultado desejado (você pode assumir que, mesmo que o resultado não seja um número inteiro, ele será mostrado mesmo assim). Arthur, porém, começou a estudar outras matérias mais avançadas, como multiplicação e geometria, e já não tem tanto tempo livre para descobrir maneiras de fazer divisões em sua calculadora. Ele pede a sua ajuda para fazer um programa que, dados o número de bits da calculadora, o dividendo e o divisor, diga a ele a melhor maneira de se calcular a divisão em tal calculadora. A melhor maneira é aquela em que o dividendo e o divisor são os menores possíveis, e ainda assim o resultado é exatamente o mesmo que o da divisão dada na entrada.

Você deve fazer ma função chamada **mini_calculadora** que recebe três argumentos. O primeiro argumento é o número de bits da calculadora. O segundo argumento é o dividendo da conta que Arthur precisa fazer. O terceiro argumento é o divisor da conta que Arthur precisa fazer.

Sua função deve retornar uma tupla de dois inteiros R e P , onde R é o dividendo e P é o divisor, mostrando a melhor maneira possível de se efetuar a divisão na calculadora de Arthur. Se for impossível realizar essa divisão na calculadora dada, a sua função deve retornar a string 'IMPOSSIVEL' (com letras maiúsculas, sem acento).

Exemplos:

Chamada da função	Retorno da função
<code>mini_calculadora(30,200,90)</code>	<code>(20,9)</code>
<code>mini_calculadora(5,62,5)</code>	<code>'IMPOSSIVEL'</code>
<code>mini_calculadora(100,31,29)</code>	<code>(31,29)</code>