



OBI2008

Caderno de Tarefas

Modalidade Programação • Nível Júnior, Fase 2
26 de Abril de 2008

A PROVA TEM DURAÇÃO DE TRÊS HORAS

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 5 páginas (não contando esta folha de rosto), numeradas de 1 a 5. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Pascal devem ser arquivos com sufixo *.pas*. Para problemas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada problema.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou disquete, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
 - em Pascal: *readln*, *read*, *writeln*, *write*;
 - em C: *scanf*, *getchar*, *printf*, *putchar*;
 - em C++: as mesmas de C ou os objetos *cout* e *cin*.
- Procure resolver o problema de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

Sociedade Brasileira de Computação

www.sbc.org.br

Fundação Carlos Chagas

www.fcc.org.br

Auto Estrada

Nome do arquivo fonte: `auto.c`, `auto.cpp` ou `auto.pas`

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.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha contém um inteiro C ($1 \leq C \leq 10^6$), indicando o comprimento da auto estrada, em blocos de 10 metros. A linha seguinte contém C caracteres, cada letra descrevendo um bloco de 10 metros da auto estrada, como definido acima.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha contendo um número inteiro, representando quantas unidades de painel são necessárias para cobrir toda a extensão da auto estrada.

Exemplo de entrada	Exemplo de saída
5 DAPCD	5

Exemplo de entrada	Exemplo de saída
8 AACCAAPP	12

Exemplo de entrada	Exemplo de saída
14 ADCCPPPPAADCP	21

Informações sobre a pontuação

- Para um subconjunto dos casos de teste totalizando 30 pontos, $1 \leq C \leq 100$.
- Para um subconjunto dos casos de teste totalizando 55 pontos, $1 \leq C \leq 10^3$.

Mini Calculadora

Nome do arquivo fonte: `minicalc.c`, `minicalc.cpp` ou `minicalc.pas`

Arthur é um menino pobre, e por isso tudo o que ele tem é de qualidade inferior. Mas 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 – ela reconhece apenas números pequenos.

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). 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 supor que, mesmo que o resultado não seja um número inteiro, ele será mostrado pela calculadora).

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 valor máximo que sua calculadora consegue representar, o dividendo e o divisor, determina a melhor maneira de se calcular a divisão em sua calculadora. Note que a melhor maneira é aquela em que o dividendo e o divisor são os menores possíveis, podem ser representados na calculadora e o resultado é exatamente o mesmo que o da divisão desejada.

Entrada

A entrada contém um único teste, a ser lido da *entrada padrão*. O teste contém uma linha com três inteiros C , D , Q ($1 \leq C \leq 1000$, $1 \leq D \leq 1000$, $2 \leq Q \leq 1000$) indicando, respectivamente, o valor máximo que pode ser representado na calculadora, o dividendo e o divisor da conta que Arthur precisa fazer.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, mostrando a melhor maneira possível de se efetuar a divisão na calculadora de Arthur. A linha deve conter dois números inteiros R e P , separados por um espaço em branco, onde R é o novo dividendo e P é o novo divisor. Se for impossível realizar essa divisão na calculadora dada, imprima a palavra `IMPOSSIVEL` (maiúsculas, sem acento).

Exemplo de entrada	Exemplo de saída
30 200 90	20 9

Exemplo de entrada	Exemplo de saída
16 62 58	IMPOSSIVEL

Exemplo de entrada	Exemplo de saída
100 31 29	31 29

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos, $C \leq 10$, $D \leq 10$ e $Q \leq 10$.
- Em um conjunto de casos de teste que totaliza 55 pontos, $D \leq 100$, $D \leq 100$ e $Q \leq 100$.

Ações da Bolsa

Nome do arquivo fonte: `acoes.c`, `acoes.cpp` ou `acoes.pas`

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 conseqüentemente 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 $N \geq 4$.

Entrada

A entrada contém um único teste, a ser lido da *entrada padrão*. O teste contém duas linhas. Na primeira, é dado um número inteiro N ($4 \leq N \leq 100000$), indicando a quantidade de dias que Juninho já sabe qual será a variação do valor da ação. Na segunda linha são dados N números inteiros X_i ($-1000 \leq X_i \leq 1000$), separados por espaços em branco, sendo X_i a variação do preço das ações no dia i .

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo 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.

Exemplo de entrada	Exemplo de saída
5 3 1 -2 -3 5	1

Exemplo de entrada	Exemplo de saída
6 -2 1 -2 1 0 -3	0

Exemplo de entrada	Exemplo de saída
8 -1 -2 -3 -4 -5 -6 -7 -8	-10

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos, $N \leq 10$ e $0 \leq X_i \leq 100$.
- Em um conjunto de casos de teste que totaliza 55 pontos, $N \leq 100$ e $-100 \leq X_i \leq 100$.