



OBI2017

Caderno de Tarefas

Modalidade Programação • Nível 1 • Fase 1

12 de maio de 2017

A PROVA TEM DURAÇÃO DE 2 HORAS

Promoção:



Sociedade Brasileira de Computação

Apoio:



Instruções

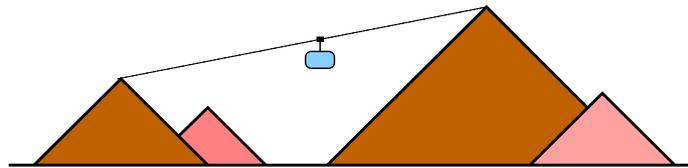
LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 3 páginas (não contando a folha de rosto), numeradas de 1 a 3. 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*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python devem ser arquivos com sufixo *.py2* para python2 e *.py3* para python3; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*. 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 pen-drive, 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*.
 - em Java: qualquer classe ou função padrão, como por exemplo *Scanner, BufferedReader, BufferedWriter* e *System.out.println*
 - em Python: *read, readline, readlines, input, print, write*
 - em Javascript: *scanf, printf*
- 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.

Teleférico

Nome do arquivo: `teleferico.c`, `teleferico.cpp`, `teleferico.pas`, `teleferico.java`,
`teleferico.js` ou `teleferico.py`

A turma do colégio vai fazer uma excursão na serra e todos os alunos e monitores vão tomar um teleférico para subir até o pico de uma montanha. A cabine do teleférico pode levar C pessoas no máximo, contando alunos e monitores, durante uma viagem até o pico. Por questão de segurança, tem que ter pelo menos um monitor dentro da cabine junto com os alunos. Por exemplo, se cabem $C = 10$ pessoas na cabine e a turma tem $A = 20$ alunos, o colégio poderia fazer três viagens: a primeira com 8 alunos e um monitor; a segunda com 6 alunos e um monitor; e a terceira com 6 alunos e um monitor. Você consegue ver que não seria possível fazer apenas duas viagens?



Dados como entrada a capacidade C da cabine e o número total A de alunos, você deve escrever um programa para calcular o número mínimo de viagens do teleférico.

Entrada

A primeira linha da entrada contém um inteiro C , representando a capacidade da cabine. A segunda linha da entrada contém um inteiro A , representando o número total de alunos na turma.

Saída

Seu programa deve imprimir uma linha contendo um número inteiro representando o número mínimo de viagens do teleférico para levar todos os alunos até o pico da montanha.

Restrições

- $2 \leq C \leq 100$ e $1 \leq A \leq 1000$

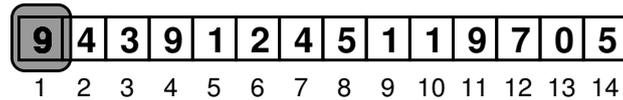
Exemplos

Entrada 10 20	Saída 3
Entrada 12 55	Saída 5
Entrada 100 87	Saída 1

Segredo do Cofre

Nome do arquivo: `cofre.c`, `cofre.cpp`, `cofre.pas`, `cofre.java`, `cofre.js` ou `cofre.py`

O sistema de segredo para abrir esse cofre é bastante complexo. Ao invés de girar um botão várias vezes, como a gente vê normalmente nos filmes, o dono do cofre tem que deslizar um controle para a esquerda e para a direita, em cima de uma barra, várias vezes, parando em determinadas posições. A barra possui N posições e cada posição contém um número inteiro entre 0 e 9, inclusive. No exemplo da figura, a barra tem 14 posições e o controle está na posição 1.



O segredo vai depender de quantas vezes cada um dos dez inteiros entre 0 e 9 vai aparecer dentro do controle. Por exemplo, suponha que o dono deslize o controle da posição inicial 1 até a posição 9, depois para a posição 4, depois para a posição 11 e por fim até a posição 13. Veja que o inteiro 1, por exemplo, vai aparecer seis vezes dentro do controle; e o inteiro 9 vai aparecer quatro vezes.

Dada a sequência de inteiros na barra e a sequência de posições entre as quais o dono desliza o controle, começando da posição inicial 1, seu programa deve contar quantas vezes cada inteiro, entre 0 e 9, vai aparecer dentro do controle.

Entrada

A primeira linha da entrada contém dois inteiros N e M , representando o número de posições na barra do cofre e o número de posições na sequência que o dono vai seguir para deslizar o controle. A segunda linha contém N inteiros entre 0 e 9, definindo a barra do cofre. A terceira linha contém M inteiros representando a sequência de posições que o dono vai seguir. A primeira posição nessa sequência é sempre 1 e não há duas posições consecutivas iguais.

Saída

Seu programa deve imprimir uma linha contendo 10 inteiros, representando o número de vezes que cada inteiro, entre 0 e 9, vai aparecer no controle da barra.

Restrições

- $2 \leq N \leq 10^5$ e $2 \leq M \leq 10^5$

Informações sobre a pontuação

- Em um conjunto de testes somando 40 pontos, $N \leq 1000$ e $M \leq 1000$

Exemplos

<p>Entrada</p> <pre>14 5 9 4 3 9 1 2 4 5 1 1 9 7 0 5 1 9 4 11 13</pre>	<p>Saída</p> <pre>1 6 3 1 4 3 0 1 0 4</pre>
<p>Entrada</p> <pre>5 4 5 8 0 5 1 1 4 2 5</pre>	<p>Saída</p> <pre>3 1 0 0 0 3 0 0 2 0</pre>