



**OBI2010**

## **Caderno de Tarefas**

**Modalidade Programação • Nível 2, Fase 2**

24 de abril de 2010

**A PROVA TEM DURAÇÃO DE 5 HORAS**

**Promoção:**



**Sociedade Brasileira de Computação**

**Patrocínio:**



**Fundação Carlos Chagas**

# Instruções

## LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 10 páginas (não contando esta folha de rosto), numeradas de 1 a 10. 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.

# Floresta

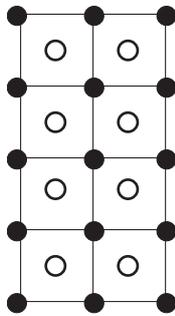
Nome do arquivo fonte: floresta.c, floresta.cpp, ou floresta.pas

O desmatamento é um dos maiores problemas enfrentados pelo Brasil hoje; estima-se que mais de 10 mil km<sup>2</sup> de vegetação sejam desflorestados todo ano. Além de destruir os habitats de várias espécies em risco de extinção, o desmatamento promove a emissão de gás carbônico, principal responsável pelo efeito estufa e pelo aquecimento global.

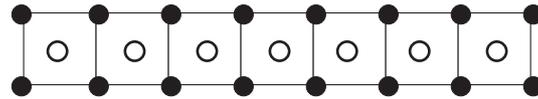
A Fundação de Conservação dos Carvalhos (FCC) tenta combater esta tendência, promovendo o reflorestamento das regiões desmatadas. Para isso, eles pretendem plantar carvalhos formando um quadriculado (um carvalho em cada vértice); no centro de cada quadrado formado por eles, a FCC também plantará um eucalipto. Para preservar a biodiversidade da área plantada, pelo menos uma árvore de cada espécie deve ser plantada durante o reflorestamento.

Por exemplo, se a FCC quiser plantar 23 árvores, ela poderá fazê-lo de duas maneiras: ou formando um retângulo  $3 \times 5$  com os carvalhos, como na figura (a), ou formando um retângulo  $2 \times 8$ , como na figura (b).

Considere que, para os propósitos deste problema, um retângulo  $x \times y$  é equivalente a um retângulo  $y \times x$ .



(a)



(b)

## Tarefa

Escreva um programa que, dado o número total de árvores que devem ser plantadas, de quantas maneiras diferentes elas podem ser dispostas.

## Entrada

A única linha da entrada contém um único inteiro  $N$ , que indica o número total de árvores que devem ser plantadas ( $1 \leq N \leq 10^9$ ).

## Saída

Seu programa deve imprimir uma única linha, contendo um único inteiro, indicando o número de arranjos distintos que podem ser feitos para o reflorestamento.

## Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 40 pontos,  $N \leq 10.000$ .

## Exemplos

<b>Entrada</b>	<b>Saída</b>
23	2

<b>Entrada</b>	<b>Saída</b>
7	0

<b>Entrada</b>	<b>Saída</b>
53	3

# Altas Aventuras

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

Incentivado por um filme de animação recente, vovô resolveu realizar seu sonho de criança, fazendo sua pequena casa voar amarrada a balões de hélio. Comprou alguns balões coloridos de boa qualidade, para fazer alguns testes, e começou a planejar a grande aventura. A primeira tarefa é determinar qual a quantidade de hélio máxima que pode ser injetada em cada balão de maneira que ele não estoure.

Suponha que os valores possíveis de quantidade de hélio em cada balão variem entre os valores 1 e  $N$ . Claro que vovô poderia testar todas as possibilidades, mas esse tipo de solução ineficiente não é apropriada, ainda mais considerando que vovô comprou apenas  $K$  balões para os testes.

Por exemplo, suponha que  $N = 5$  e  $K = 2$ . Nesse caso, a melhor solução seria testar primeiro em 3. Caso o balão estoure, vovô só teria mais um balão, então teria de testar 1 e 2 no pior caso, somando ao todo 3 testes. Caso o balão não estoure, vovô poderia testar 4 e depois 5 (ou 5 e depois 4), também somando 3 ao todo.

## Tarefa

Dados a capacidade máxima da bomba e o número de balões, indicar o número mínimo de testes que devem ser feitos, no pior caso, para determinar o ponto em que um balão estoura.

## Entrada

A única linha da entrada contém dois inteiros,  $N$  e  $K$ , separados por espaço em branco ( $1 \leq K \leq N \leq 1.000.000.000$ ).

## Saída

Seu programa deve imprimir uma única linha, contendo um inteiro que representa o número mínimo de testes que devem ser feitos no pior caso para determinar o ponto em que o balão estoura.

## Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 40 pontos, ( $1 \leq K \leq N \leq 200$ ).

## Exemplos

<b>Entrada</b> 5 2	<b>Saída</b> 3
<b>Entrada</b> 20 2	<b>Saída</b> 6
<b>Entrada</b> 11 5	<b>Saída</b> 4

# Tradutor alienígena

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

É de conhecimento público e notório que já fomos visitados por alienígenas diversas vezes. A grande dificuldade que temos, porém, é a comunicação com eles, por causa de grandes diferenças entre as línguas. Além disso, assim como nós, eles também têm várias línguas diferentes.

Com o intuito de auxiliar no processo de tradução, foi criado um método de mapeamento dos símbolos do alfabeto de cada língua alienígena, atribuindo um número inteiro para cada símbolo. Sendo assim, para um alfabeto alienígena com  $N$  elementos, atribui-se números de 1 a  $N$  a cada um.

O problema é que o encarregado de transcrever os textos alienígenas para números não foi muito cuidadoso e usou o mesmo espaçamento entre dígitos e números. Assim, por exemplo, digamos que para um alfabeto com 32 símbolos, uma sequência que deveria ser “31 20 4 19” virou “3120419”. Como se pode notar, há diferentes maneiras válidas de interpretar essa sequência além da original, como por exemplo “3 1 20 4 19” e “31 20 4 1 9”. Repare que a transcrição nunca usa zeros à esquerda de um número e, portanto, a sequência “3 12 04 19” é inválida, assim como “31 20 41 9” por conter um número (49) que não corresponde a um símbolo.

## Tarefa

Dados a quantidade de símbolos do alfabeto e uma sequência transcrita, determine quantas sequências válidas podem ser formadas.

## Entrada

A entrada é composta por duas linhas. A primeira contém um número inteiro  $N$  ( $1 < N < 10^{100}$ ) que indica a quantidade de símbolos do alfabeto. A segunda linha contém uma cadeia de dígitos de tamanho mínimo 1 e tamanho máximo 100.000 que corresponde a sequência transcrita.

## Saída

Seu programa deve imprimir uma linha com o resto da divisão da quantidade de sequências válidas por 1.000.000.007.

## Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos, a cadeia de caracteres tem no máximo 20 dígitos e  $N < 100$ .

## Exemplos

<b>Entrada</b>	<b>Saída</b>
32 3120419	4

<b>Entrada</b>	<b>Saída</b>
32 4021333251231253	0

<b>Entrada</b>	<b>Saída</b>
500 12345	13

# Multiplicação de matrizes

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

O conglomerado indiano Tutu é um conjunto de empresas que atua nos mais diversos ramos da indústria, produzindo desde sapatos até aviões e foguetes. Por ser tão diversificada, precisa de grandes e rápidos sistemas para cálculos de contabilidade.

Um dos módulos mais importantes desse sistema é o de fornecimento de produtos, onde fica a base de dados de produtos e fornecedores. Um mesmo produto pode ser fornecido por vários fornecedores diferentes.

O sistema possui duas grandes matrizes: a matriz **A**, onde cada linha representa um produto e cada coluna representa um fornecedor. O valor da matriz na linha  $m$  e coluna  $n$  representa o preço do produto  $m$  se for comprado do fornecedor  $n$ .

A outra grande matriz é a **B**, onde cada linha representa um dia do mês e cada coluna é um produto. O valor da matriz na linha  $m$  e coluna  $n$  representa a quantidade do produto  $n$  a ser adquirido no dia  $m$ .

Tal empresa tem uma política de fidelidade com seus fornecedores, e uma das práticas efetuadas pela empresa é, em um determinado dia, comprar todos os produtos necessários de um único fornecedor. Isto é, em um dia todos os produtos adquiridos serão comprados do fornecedor  $x$ , no outro dia do fornecedor  $y$ , e assim por diante.

Para auxiliar a escolha de qual fornecedor será o escolhido no dia, foi gerada outra matriz **C**, que é o resultado da multiplicação das matrizes  $\mathbf{A} \times \mathbf{B}$ . Essa matriz diz o quanto será gasto pela empresa se adquirir todos os produtos de um determinado fornecedor em um determinado dia.

As matrizes **A** e **B** são quadradas (o número de linhas é igual ao número de colunas) e têm valores definidos pelas fórmulas

$$\mathbf{A}_{ij} = (P \times i + Q \times j) \pmod{X}$$

$$\mathbf{B}_{ij} = (R \times i + S \times j) \pmod{Y}$$

onde  $i$  é o índice da linha da matriz e  $j$  é o índice da coluna da matriz (todos os índices vão de 1 até  $N$ ). Os inteiros  $P$ ,  $Q$ ,  $R$ ,  $S$ ,  $X$  e  $Y$  são parâmetros constantes, que definem as duas matrizes **A** e **B**.

## Tarefa

Escreva um programa que, dados os parâmetros das matrizes **A** e **B**, e a posição de uma das entradas da matriz **C**, calcula o valor daquela entrada.

## Entrada

A primeira linha da entrada contém um inteiro  $N$ , indicando as dimensões das matrizes **A**, **B** e **C** ( $2 \leq N \leq 10^5$ ). A linha seguinte contém seis inteiros  $P$ ,  $Q$ ,  $R$ ,  $S$ ,  $X$  e  $Y$ , indicando os parâmetros das matrizes **A** e **B** ( $2 \leq X, Y \leq 10^4$ ;  $0 \leq P, Q < X$ ;  $0 \leq R, S < Y$ ). Finalmente, a última linha da entrada contém dois inteiros  $I$  e  $J$ , indicando a linha e a coluna da matriz **C** a serem consultados ( $1 \leq I, J \leq N$ ).

## Saída

Seu programa deve imprimir uma única linha contendo o valor da matriz **C** na linha e coluna especificadas.

## Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 40 pontos,  $N \leq 100$ .

## Exemplos

<b>Entrada</b>	<b>Saída</b>
3 4 3 2 3 5 6 2 2	18

<b>Entrada</b>	<b>Saída</b>
4 3 5 1 0 6 7 4 3	30

<b>Entrada</b>	<b>Saída</b>
2 2 2 0 1 3 2 2 1	2

# Telescópio

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

Telescópios são instrumentos que auxiliam a observação do céu, melhorando e aumentando o aspecto das estrelas, planetas e outros objetos brilhantes. Existem diversos tipos de telescópios, sendo os tipos mais comuns os de lentes objetivas (refratores) e os de espelhos (refletores).

A maneira como os telescópios melhoram a nossa percepção dos astros no céu é aumentando a quantidade de luz captada que chega aos nossos olhos. Toda luz que entra pelos nossos olhos entra por um orifício chamado pupila. Tal controla a quantidade de luz que entra nos olhos, aumentando o diâmetro quando o ambiente está escuro (e portanto precisamos obter mais luz para identificar os objetos) e diminuindo quando o ambiente está claro. Num ambiente muito escuro, a pupila pode atingir um diâmetro de 8 mm.

Cada objeto celeste (estrela, planeta, nebulosa, etc) emite uma quantidade de luz (fótons) que é homogeneamente distribuída quando chega na Terra. Por exemplo, a estrela  $A$  emite luz que pode ser captada a um fluxo de 40.000 fótons por segundo por milímetro quadrado. Isso é, a cada segundo, é possível captar 40.000 fótons provenientes da estrela  $A$  numa área de  $1 \text{ mm}^2$ . Ou seja, uma pupila de  $10 \text{ mm}^2$  de área captaria 400.000 fótons provenientes da estrela  $A$  por segundo.

Para que nosso cérebro consiga interpretar que existe um objeto ali, porém, ele precisa receber 40.000.000 fótons por segundo. Assim, podemos utilizar um telescópio com lente (ou espelho) de  $100 \text{ mm}^2$  de área, que vai captar a quantidade necessária de fótons provenientes da estrela  $A$  e encaminhá-los até nossa pupila, fazendo assim com que nosso cérebro perceba a presença da estrela ali.

## Tarefa

Dada uma lista com estrelas no céu, o fluxo de fótons que cada uma delas emite, e área de abertura de um telescópio, dizer quantas estrelas serão perceptíveis usando tal telescópio.

## Entrada

A primeira linha da entrada terá um inteiro  $A$  ( $1 \leq A \leq 10.000$ ) representando a área de abertura do telescópio (em milímetros quadrados) a ser considerado. A segunda linha possui um inteiro  $N$  ( $1 \leq N \leq 10.000$ ) representando o número de estrelas a serem estudadas. As  $N$  linhas seguintes terão, cada uma, um inteiro  $F$  ( $1 \leq F \leq 20.000$ ) representando o fluxo de fótons que cada uma das  $N$  estrelas emitem (em fótons por segundo por milímetro quadrado).

## Saída

Imprima um inteiro representando a quantidade de estrelas que serão percebidas ao se utilizar o telescópio em questão.

## Informações sobre a pontuação

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

## Exemplos

<b>Entrada</b>	<b>Saída</b>
10000 3 4000 3500 5100	2

<b>Entrada</b>	<b>Saída</b>
5869 3 3975 14234 8569	2

<b>Entrada</b>	<b>Saída</b>
2967 9 18650 16338 2400 17702 14619 13934 7979 16316 10533	6