

Competidor(a): \_\_\_\_\_

Número de inscrição: \_\_\_\_\_ – \_\_\_\_\_ (opcional)

*Este Caderno de Tarefas não pode ser levado para casa após a prova. Após a prova entregue este Caderno de Tarefas para seu professor guardar. Os professores poderão devolver os Cadernos de Tarefas aos competidores após o término do período de aplicação das provas (01 de setembro de 2023).*



Olimpíada Brasileira de Informática  
Competição Feminina - OBI2023

Caderno de Tarefas  
Modalidade Programação • Nível 2 • Fase Única

01 de setembro de 2023

A PROVA TEM DURAÇÃO DE 3 HORAS

Promoção:



Sociedade Brasileira de Computação

Apoio:



Coordenação:



# Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 8 páginas (não contando a folha de rosto), numeradas de 1 a 8. 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; em particular, seu programa não deve escrever frases como “Digite o dado de entrada:” ou similares.
- 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 necessariamente 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 Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 3 devem ser arquivos com sufixo *.py*; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*.
- Na linguagem Java, **não** use o comando *package*, e note que o nome de sua classe principal deve usar somente letras minúsculas (o mesmo nome do arquivo indicado nas tarefas).
- Você pode submeter até 50 soluções para cada tarefa. A pontuação total de cada tarefa é a melhor pontuação entre todas as submissões. Se a tarefa tem sub-tarefas, para cada sub-tarefa é considerada a melhor pontuação entre todas as submissões.
- 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 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 a tarefa 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.

# Estante de Livros

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

A princesa Jujuba encontrou 3 pilhas em uma sala secreta de seu castelo. A primeira pilha tem  $X$  livros, a segunda pilha tem  $Y$  livros e a terceira pilha tem  $Z$  livros. Jujuba quer colocar seus livros na sua estante, que tem  $N$  prateleiras.

Cada prateleira pode conter infinitos livros, mas como ela é organizada, é preciso que todas as prateleiras tenham a mesma quantidade de livros. Como nem sempre isso é possível, Jujuba quer saber qual a menor quantidade de livros  $L$  que terão de ficar fora da sua estante.

Um exemplo seria as pilhas terem 2, 5 e 9 livros e sua estante ter 3 prateleiras. Jujuba poderia colocar 1 livro em cada prateleira, deixando 13 livros de fora. Mas a melhor resposta seria Jujuba colocar 5 livros em cada prateleira, deixando apenas 1 livro de fora.

Você consegue ajudar a princesa Jujuba?

## Entrada

A única linha de entrada contém quatro inteiros  $X$ ,  $Y$ ,  $Z$  e  $N$ , descritos no enunciado.

## Saída

Seu programa deve produzir uma única linha contendo um único inteiro  $L$ .

## Restrições

- $1 \leq X, Y, Z, N \leq 10^{18}$

## Informações sobre a pontuação

- $X, Y, Z, N \leq 10^6$  (73 pontos)
- Sem mais restrições (27 pontos)

## Exemplos

|  |                                |
|--|--------------------------------|
| <b>Exemplo de entrada 1</b><br>2 5 9 3 | <b>Exemplo de saída 1</b><br>1 |
| <b>Exemplo de entrada 2</b><br>1 2 3 4 | <b>Exemplo de saída 2</b><br>2 |

# Machine Learning

*Nome do arquivo:* `machine-learning.c`, `machine-learning.cpp`, `machine-learning.java`,  
`machine-learning.js` ou `machine-learning.py`

Leticia está fazendo faculdade de Ciências da Computação e está muito animada para começar a estudar sobre Aprendizado de Máquina. Porém, ela ainda está no segundo semestre da faculdade, então ainda irá demorar um tempo para ter aulas sobre o assunto. Ela decidiu então construir um modelo de aprendizado de máquina mais simples, com os conhecimentos que ela possui de programação até então.

Sua ideia foi a seguinte: ela irá construir um programa que, primeiramente, lê vários tópicos de artigos e palavras relacionadas a esses tópicos. Depois, o programa irá ler o texto de um artigo e irá definir qual o seu tópico. O tópico do artigo será aquele que possui mais palavras relacionadas ao tópico mencionadas no artigo.

Se houver empate, o tópico do artigo será aquele que antecede o outro no alfabeto. Uma palavra será relacionada a no máximo um tópico. É garantido que o artigo terá pelo menos uma palavra relacionada a algum tópico.

Leticia pediu sua ajuda para desenvolver o algoritmo.

## Entrada

A primeira linha de entrada contém um inteiro  $N$ , o número de tópicos possíveis. As próximas  $N$  linhas são compostas cada uma por uma string  $S_i$ , o nome do tópico, seguida de um inteiro  $K_i$ , o número de palavras relacionadas ao tópico, seguido por  $K_i$  strings  $s_j$ , as palavras relacionadas ao tópico  $S_i$ . A próxima linha contém um inteiro  $X$ , o número de palavras do artigo. A última linha contém  $X$  strings  $x_l$ , as palavras do artigo.

## Saída

Seu programa deve produzir uma única linha contendo uma única palavra, o nome do tópico do artigo.

## Restrições

- $1 \leq N \leq 100$
- $1 \leq$  número de caracteres de  $S_i \leq 20$
- $1 \leq K_i \leq 100$
- $1 \leq$  número de caracteres de  $s_j \leq 20$
- $1 \leq X \leq 10^5$
- $1 \leq$  número de caracteres de  $x_l \leq 20$
- Todas as strings serão compostas apenas de letras minúsculas do alfabeto

## Informações sobre a pontuação

- $N = 1$  (18 pontos)
- $X \leq 10^3$  (28 pontos)
- Sem mais restrições (54 pontos)

## Exemplos

|  |                                       |
|--|---------------------------------------|
| <b>Exemplo de entrada 1</b><br>3<br>ciencia 3 quimica fisica matematica<br>amor 4 coracao amo carinho afeto<br>livros 3 paginas paragrafo textos<br>7<br>eu amo textos sobre quimica e fisica  | <b>Exemplo de saída 1</b><br>ciencia  |
| <b>Exemplo de entrada 2</b><br>4<br>tecnologia 2 software desenvolvimento<br>saude 3 alimentacao exercicio vacina<br>financas 4 boleto moeda contas cartao<br>internet 3 site youtube instagram<br>12<br>eu assisto videos sobre desenvolvimento de software no instagram e no youtube | <b>Exemplo de saída 2</b><br>internet |

# Viagem

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

Maria mora na Nlogônia, um país composto por  $N$  cidades conectadas por  $N - 1$  estradas bidirecionais de tal forma que é sempre possível viajar entre quaisquer duas cidades.

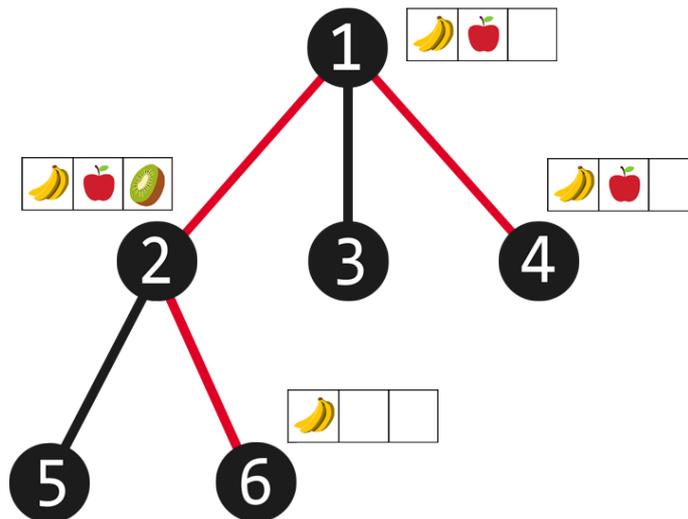
O solo da Nlogônia é muito fértil e permite o cultivo de  $K$  diferentes tipos de frutas onde cada cidade cultiva um conjunto específico desses tipos de frutas (podendo ser todos os tipos ou nenhum).

O comércio da Nlogônia é sua principal fonte de renda, sendo assim todas as cidades têm um grande mercado para vender todos os tipos de frutas que aquela cidade cultiva.

Quando Maria faz uma viagem entre duas cidades ela sempre vai pelo menor caminho e tenta comprar todos os tipos de frutas. Maria comprará todos os tipos que aparecem uma quantidade ímpar de vezes no trajeto de sua viagem, caso contrário, ela não comprará.

Maria está com a grana curta e pediu a sua ajuda para descobrir quantos tipos de frutas ela comprará em cada uma de suas  $Q$  viagens.

Confira um exemplo com 6 cidades, 3 tipos de frutas em uma viagem da cidade 6 para a cidade 4.



A viagem feita por Maria está destacada em vermelho. Nesse trajeto, a banana aparece 4 vezes (mercados das cidades 1, 2, 4 e 6), a maçã 3 vezes (mercados das cidades 1, 2 e 4) e o kiwi 1 vez (mercado da cidade 2), sendo assim Maria comprará apenas maçãs e kiwis, totalizando 2 tipos comprados em sua viagem.

Dado a informação do país de Maria, de suas viagens e dos mercados em cada cidade, ajude Maria a responder quantos tipos de frutas ela comprará em cada viagem.

## Entrada

A primeira linha contém dois inteiros  $N$  e  $K$ , o número de cidades no país de Maria e quantos tipos de frutas são cultivadas lá, respectivamente.

As  $N - 1$  linhas seguintes contém dois inteiros cada  $X$  e  $Y$ , indicando que existe uma estrada que liga as cidades  $X$  e  $Y$ .

As próximas linhas da entrada descrevem uma matriz booleana  $M_{N \times K}$ , onde  $m_{i,j} = 0$  ou  $1$ , indicando se o mercado da cidade  $i$  vende o tipo de fruta  $j$ .

A próxima linha contém um inteiro  $Q$ , o número de viagens que Maria vai fazer. As  $Q$  linhas seguintes contém dois inteiros cada  $X$  e  $Y$ , a origem e o destino da viagem de Maria respectivamente.

### Saída

Seu programa deve produzir  $Q$  linhas, cada uma com um único inteiro, respondendo quantos tipos de frutas maria vai comprar em sua  $i$ -ésima ( $i = 1, 2, \dots, Q$ ) viagem.

### Restrições

- $1 \leq N, Q \leq 10^5$
- $1 \leq K \leq 64$

### Informações sobre a pontuação

- $N, Q \leq 10^3$  e  $K = 1$  (19 pontos)
- $N, Q \leq 10^3$  (18 pontos)
- $K = 1$  (29 pontos)
- Sem mais restrições (34 pontos)

### Exemplos

| Exemplo de entrada 1 | Exemplo de saída 1 |
|----------------------|--------------------|
| 6 3                  | 2                  |
| 1 2                  | 2                  |
| 1 3                  | 1                  |
| 1 4                  | 0                  |
| 2 5                  | 0                  |
| 2 6                  |                    |
| 1 1 0                |                    |
| 1 1 1                |                    |
| 0 0 0                |                    |
| 1 1 0                |                    |
| 0 1 0                |                    |
| 1 0 0                |                    |
| 5                    |                    |
| 6 4                  |                    |
| 5 3                  |                    |
| 5 6                  |                    |
| 4 1                  |                    |
| 3 3                  |                    |

*Explicação do exemplo 1:*

Contagem de frutas em cada viagem

|          | Tipo 1 | Tipo 2 | Tipo 3 | Comprou     |
|----------|--------|--------|--------|-------------|
| Viagem 1 | 4      | 3      | 1      | Tipos 2 e 3 |
| Viagem 2 | 2      | 3      | 1      | Tipos 2 e 3 |
| Viagem 3 | 2      | 2      | 1      | Tipo 3      |
| Viagem 4 | 2      | 2      | 3      | Nenhum tipo |
| Viagem 5 | 0      | 0      | 0      | Nenhum tipo |

| Exemplo de entrada 2 | Exemplo de saída 2 |
|----------------------|--------------------|
| 7 3                  | 1                  |
| 1 2                  | 2                  |
| 1 3                  | 3                  |
| 2 4                  | 2                  |
| 2 5                  | 2                  |
| 3 6                  |                    |
| 3 7                  |                    |
| 1 1 1                |                    |
| 1 0 1                |                    |
| 0 1 0                |                    |
| 1 1 0                |                    |
| 1 0 1                |                    |
| 1 1 0                |                    |
| 0 1 1                |                    |
| 5                    |                    |
| 1 2                  |                    |
| 4 5                  |                    |
| 1 5                  |                    |
| 4 7                  |                    |
| 6 6                  |                    |

*Explicação do exemplo 2:*

Contagem de frutas em cada viagem

|          | Tipo 1 | Tipo 2 | Tipo 3 | Comprou          |
|----------|--------|--------|--------|------------------|
| Viagem 1 | 2      | 1      | 2      | Tipo 2           |
| Viagem 2 | 3      | 1      | 2      | Tipos 1 e 2      |
| Viagem 3 | 3      | 1      | 3      | Todos os 3 tipos |
| Viagem 4 | 3      | 4      | 3      | Tipos 1 e 3      |
| Viagem 5 | 1      | 1      | 0      | Tipos 1 e 2      |

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

*Explicação do exemplo 3:* Não existem frutas à venda em toda Nlogônia ):

| Exemplo de entrada 4 | Exemplo de saída 4 |
|----------------------|--------------------|
| 3 10                 | 10                 |
| 1 2                  | 10                 |
| 2 3                  | 0                  |
| 1 1 1 1 1 1 1 1 1 1  |                    |
| 0 0 0 0 0 0 0 0 0 0  |                    |
| 1 1 1 1 1 1 1 1 1 1  |                    |
| 3                    |                    |
| 1 2                  |                    |
| 2 3                  |                    |
| 1 3                  |                    |

*Explicação do exemplo 4:*

Contagem de frutas em cada viagem

|          | Tipo 1 | ... | Tipo 10 | Comprou           |
|----------|--------|-----|---------|-------------------|
| Viagem 1 | 1      | ... | 1       | Todos os 10 tipos |
| Viagem 2 | 0      | ... | 0       | Nenhum tipo       |
| Viagem 3 | 1      | ... | 1       | Todos os 10 tipos |