

## INSTRUÇÕES GERAIS

- A inscrição para a competição é feita através desse link, e exige uma conta no [codeforces.com](https://codeforces.com) (caso não possua, pode criar uma em [codeforces.com/register](https://codeforces.com/register)):  
[goo.gl/xeF4V6](https://goo.gl/xeF4V6)
- O caderno de questões contém **17** páginas numeradas, contando com esta.
- A prova é formada por **6** problemas, identificados por letras de **A** a **F**.
- Para cada problema, deve ser feito um programa de computador para resolvê-lo.
- Esse programa deve ler a partir da entrada padrão e escrever na saída padrão.
- Cada problema consiste em um enunciado o descrevendo, seguido das descrições exatas da entrada e da saída. Não se deve escrever nada na saída diferente do que é especificado.
- Após a descrição dos problemas, segue uma ou mais páginas com exemplos de entrada e saída, podendo conter explicações desses exemplos.
- É necessário que se registre na competição também no [codeforces.com](https://codeforces.com)
  - Para isso, acesse: [codeforces.com/contests/101796](https://codeforces.com/contests/101796)
  - Ao acessar o link, clique em "Register" -> "Register"
  - Após isso, acesse [codeforces.com/gym/101796](https://codeforces.com/gym/101796)
  - Através desse último link o caderno de provas e o placar podem ser vistos, além de que as submissões de problemas e dúvidas podem ser feitas.
- Cada submissão será automaticamente submetida a vários testes, e, caso seja aceita em todos, será considerada correta.
- O ranking será feito baseado no número de problemas resolvidos. Empates no número de problemas resolvidos são classificados pelo tempo corrigido. Ganha aquele que tem o menor tempo corrigido. O tempo corrigido do competidor é dado pela soma dos tempos corrigidos somente dos problemas corretamente resolvidos pelo competidor. O tempo corrigido de um problema é dado pelo número de minutos decorridos desde o início da competição até o momento da primeira submissão correta somado com uma penalidade de 20 minutos por submissão incorreta feita anteriormente neste problema.
- São permitidas as linguagens e compiladores padrões do site: GNU GCC 5.1.0, GNU GCC C11 5.1.0, Clang++17 Diagnostics, GNU G++ 5.1.0, GNU G++11 5.1.0, GNU G++14 6.4.0, GNU G++17 7.2.0, GNU C++17 Diagnostics (DrMemory), Microsoft Visual C++ 2010, C# Mono 5, D DMD32 v2.074.1, Go 1.8, Haskell GHC 7.8.3, Java 1.8.0\_131, Kotlin 1.1.3-2, OCaml 4.02.1, Delphi 7, Free Pascal 3, PascalABC.NET 2, Perl 5.20.1, PHP 7.0.12, Python 2.7, Python 3.6, PyPy 2.7.13 (5.9.0), PyPy 3.5.3 (5.10.0), Ruby 2.0.0p645, Rust 1.21, Scala 2.12, JavaScript V8 4.8.
- É recomendado que se teste os programas em sua máquina ou em um compilador / interpretador online antes de submeter. É possível também realizar testes em: <http://codeforces.com/gym/101796/customtest>
- A competição possui 3 horas de duração e é individual, não sendo permitido comunicação entre participantes ou com o público externo após o início da prova.
- O limite de tempo de execução de cada teste é de 2 segundos, e de memória é de 256MB.
- Em caso de dúvida, submeta no site ( [Ask a question](#) em <http://codeforces.com/gym/101796> ) ou pergunte a um organizador.

## A - Hora de dormir

O jovem Railton está tendo problemas para dormir. Ele já testou várias camas e travesseiros diferentes para melhorar a qualidade de seu sono, mas nada adiantou. Contudo, agora, ele está convencido que o problema está em seu cobertor, e decidiu que ele mesmo irá fazer um novo. Mas, antes ele precisa saber a quantidade de tecido para comprar, e, para isso, pediu sua ajuda!

Ele sabe que sua cama é retangular e possui **N** centímetros de comprimento e **M** centímetros de largura. Considerando que ele precisa exatamente da quantidade de tecido para cobrir a superfície da cama, informe a ele a área total de tecido que ele precisará comprar!

### Entrada:

A entrada é composta por uma linha contendo dois inteiros: **N** e **M** ( $50 \leq \mathbf{N}, \mathbf{M} \leq 300$ ).

### Saída:

A saída deve conter apenas um inteiro que representa a área total de tecido a ser comprado, em  $\text{cm}^2$ .

EXEMPLOS
Exemplo de <b>entrada 1</b> :
50 60
Exemplo de <b>saída 1</b> :
3000
Exemplo de <b>entrada 2</b> :
200 100
Exemplo de <b>saída 2</b> :
20000

\*Obs: No exemplo 1, a cama possui 50 centímetros de comprimento e 60 centímetros de largura, tendo uma área de 3000 cm<sup>2</sup>.

## B - Jogo

Agora que Railton está conseguindo dormir melhor, ele se tornou não só um engenheiro de dados, mas um inventor! Tanto que ele já fez seu primeiro jogo!

Você é o primeiro jogador desse jogo, mas já quer mostrar que é um bom jogador. No jogo, seu personagem tem apenas uma habilidade: força. **Inicialmente, sua força é 1.** Para aumentar (ou diminuir) sua força, você tem 5 objetos que podem ser equipados, e cada um possui um coeficiente que multiplica sua força. No jogo, você pode equipar no máximo 3 objetos (ou seja, pode equipar **0, 1, 2 ou 3 dos 5 objetos**). Sua missão é determinar qual é o máximo de força que seu personagem pode obter.

### Entrada:

A entrada é composta uma linha contendo 5 inteiros, cada um entre -10 e 10 (inclusive), representando o coeficiente de cada objeto.

### Saída:

A saída deve conter apenas um inteiro contendo a maior força que seu personagem pode obter após equipar no máximo 3 objetos.

EXEMPLOS
Exemplo de <b>entrada 1</b> :
-1 0 0 2 3
Exemplo de <b>saída 1</b> :
6
Exemplo de <b>entrada 2</b> :
3 2 -4 9 7
Exemplo de <b>saída 2</b> :
189
Exemplo de <b>entrada 3</b> :
-2 -2 -2 -2 -2
Exemplo de <b>saída 3</b> :
4

\*Obs: No exemplo 1, é possível obter força igual 6 escolhendo o quarto e o quinto objetos. No exemplo 2, é possível obter força igual 189 escolhendo o primeiro, o quarto e o quinto objetos. No exemplo 3 é possível obter força igual a 4 escolhendo quaisquer dois objetos. Em todos esses exemplos, é impossível obter mais força escolhendo entre 0 e 3 objetos.

## C - Estágio

Estágios em empresas costumam exigir conhecimentos muito variados. Hélio, que acabou de entrar em uma empresa têxtil, não levou esse fato em consideração e acabou não estudando, por exemplo, matrizes. Entretanto, recentemente a empresa recebeu uma solicitação de Raiton para fabricar mais tecido para seu cobertor. Como Railton está bem exigente, esse tecido precisa seguir um padrão bem especial.

As cores do tecido podem ser vistos como números, de 1 a 9, de forma que o tecido inteiro seja representado por uma matriz desses números. Para que o tecido seja especial, ele precisa ser composto por várias réplicas de uma matriz  $P$  específica. Mais precisamente, ela deve ser repetida  $X \times Y$  vezes, ou seja, devem haver  $X$  linhas, cada linha contendo  $Y$  matrizes iguais a matriz  $P$ .

Sabendo  $X$ ,  $Y$  e  $P$ , ajude Hélio a determinar a matriz que representa o tecido completo.

### Entrada:

A primeira linha contém dois inteiros  $N$  e  $M$ , que são as dimensões da matriz  $P$ .

As linhas seguintes descrevem a matriz  $P$  ( $N$  linhas, cada uma contendo  $M$  colunas, sendo os inteiros que constituem a matriz  $P$ , estando entre 1 e 9 cada um).

Por fim, a última linha contém os inteiros  $X$  e  $Y$ .

É garantido que  $N$ ,  $M$ ,  $X$  e  $Y$  estão entre 1 e 30 (inclusive).

### Saída:

Mostre a matriz resultante após colocar  $X$  sequências (uma embaixo da outra) de matrizes, cada sequência contendo  $Y$  cópias da matriz  $P$ .

EXEMPLOS	
Exemplo de <b>entrada 1</b> :	
2 1	
4	
5	
2 3	
Exemplo de <b>saída 1</b> :	
4 4 4	
5 5 5	
4 4 4	
5 5 5	
Exemplo de <b>entrada 2</b> :	
2 3	
1 3 8	
5 4 9	
3 2	
Exemplo de <b>saída 2</b> :	
1 3 8 1 3 8	
5 4 9 5 4 9	
1 3 8 1 3 8	
5 4 9 5 4 9	
1 3 8 1 3 8	
5 4 9 5 4 9	

\*Obs: No primeiro exemplo, a matriz **P** possui 2 linhas e 1 coluna, sendo a seguinte matriz:

$$\begin{bmatrix} 4 \\ 5 \end{bmatrix}$$

E para esse exemplo, deseja-se colocar colocar 2 “linhas” contendo 3 cópias da matriz **P**:

$$\begin{bmatrix} P & P & P \\ P & P & P \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 4 \\ 5 \end{bmatrix} & \begin{bmatrix} 4 \\ 5 \end{bmatrix} & \begin{bmatrix} 4 \\ 5 \end{bmatrix} \\ \begin{bmatrix} 4 \\ 5 \end{bmatrix} & \begin{bmatrix} 4 \\ 5 \end{bmatrix} & \begin{bmatrix} 4 \\ 5 \end{bmatrix} \end{bmatrix}$$

Para o segundo exemplo, a matriz **P** possui 2 linhas e 3 colunas, sendo a seguinte matriz:

$$\begin{bmatrix} 1 & 3 & 8 \\ 5 & 4 & 9 \end{bmatrix}$$

E a matriz resultante deve ter 3 “linhas”, cada uma contendo 2 cópias da matriz **P**:

$$\begin{bmatrix} P & P \\ P & P \\ P & P \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 & 3 & 8 \\ 5 & 4 & 9 \end{bmatrix} & \begin{bmatrix} 1 & 3 & 8 \\ 5 & 4 & 9 \end{bmatrix} \\ \begin{bmatrix} 1 & 3 & 8 \\ 5 & 4 & 9 \end{bmatrix} & \begin{bmatrix} 1 & 3 & 8 \\ 5 & 4 & 9 \end{bmatrix} \\ \begin{bmatrix} 1 & 3 & 8 \\ 5 & 4 & 9 \end{bmatrix} & \begin{bmatrix} 1 & 3 & 8 \\ 5 & 4 & 9 \end{bmatrix} \end{bmatrix}$$

Note que os colchetes ( [ ] ) foram adicionados apenas para facilitar a visualização.



## D - Fogo

A empresa de Hélio acaba de pegar fogo! Ajude Hélio a sair da empresa! A empresa pode ser vista como um *grid* (tabela) de **N** linhas por **M** colunas, onde cada célula ou está vazia, de forma que Hélio pode atravessar, ou está pegando fogo, que obviamente Hélio não pode atravessar, ou é um obstáculo, que Hélio também não pode atravessar.

A cada instante de tempo, Hélio consegue se mover para uma célula adjacente que esteja **vazia**, mas, ao mesmo tempo, cada célula que está pegando fogo consegue espalhar seu fogo para **todas** suas células adjacentes, **exceto** as que possuem obstáculos. As células adjacentes são as que estão imediatamente acima, abaixo, à esquerda ou à direita da célula de referência.

Hélio pode sair do *grid* se estiver em uma célula de sua borda, isto é, que está na primeira linha, ou primeira coluna, ou última linha, ou última coluna, e, claro, se essa célula não estiver pegando fogo.

Sabendo disso, diga “Corre!” se Hélio conseguir sair do *grid* sem ser atingido pelo fogo, ou “Vish!” se é impossível que ele escape.

### Entrada:

A primeira linha contém dois inteiros **N** e **M** ( $1 \leq N, M \leq 600$ ), as dimensões do *grid*.

Depois seguem **N** linhas, cada uma com **M** caracteres, onde cada caractere é:

‘#’ para um obstáculo que nem o fogo nem Hélio podem acessar;

‘H’ para a posição inicial de Hélio;

‘F’ para uma célula que inicialmente está pegando fogo;

‘.’ (ponto) para uma célula que está inicialmente vazia.

É garantido que haverá exatamente uma célula com o caractere ‘H’.

### Saída:

Uma string contendo “Corre!” (sem aspas) se Hélio puder sair da empresa sem ser atingido pelo fogo, ou “Vish!” (sem aspas) se não for possível fazer isso.

EXEMPLOS
Exemplo de <b>entrada 1</b> :
<pre> 4 5 ##### ..H#. F.... ..... </pre>
Exemplo de <b>saída 1</b> :
Corre!
Exemplo de <b>entrada 2</b> :
<pre> 4 5 ##### ..H.. F.... ....F </pre>
Exemplo de <b>saída 2</b> :
Vish!

\*Obs: No primeiro exemplo existem várias formas de escapar. Por exemplo, Hélio pode fugir andando uma casa para baixo, e em seguida três para a direita. Dessa forma, ele sairia do *grid* sem passar por uma célula em chamas. Esse seria o *grid* com o decorrer do tempo:

```

#####      #####      #####      #####      #####
..H#.      F..#.      FF.#.      FFF#.      FFF#.
F....      FFH..      FFFH.      FFFFH      FFFFF
.....      F....      FF...      FFF..      FFFF.

```

Note que no último instante mostrado, Hélio já não se encontra mais no *grid*.

Para o segundo exemplo, qualquer caminho que leve Hélio para fora do *grid* passa por células pegando fogo ou com obstáculos, sendo então impossível escapar.

Mais exemplos na página seguinte.

EXEMPLOS
Exemplo de <b>entrada 3</b> :
<pre> 4 5 ##### ..H.. F...# ....F </pre>
Exemplo de <b>saída 3</b> :
Corre!
Exemplo de <b>entrada 4</b> :
<pre> 3 3 .#. #H# F#. </pre>
Exemplo de <b>saída 4</b> :
Vish!

\*Obs: No terceiro exemplo, há apenas uma forma de escapar, que é caso Hélio faça três movimentos para a direita. Esse seria o *grid* com o decorrer do tempo:

```

#####      #####      #####      #####
..H..      F..H.      FF..H      FFFF.
F...#      FF..#      FFFF#      FFFF#
....F      F..FF      FFFFF      FFFFF

```

Para o quarto exemplo, apesar de Hélio em nenhum momento ser atingido pelo fogo, não é possível que ele saia do *grid* devido aos obstáculos.

## E - Corrida

Apesar do jogo de Railton não ser muito popular, jogos de carros são bem populares. Tanto é que Hélio, após sobreviver ao problema D, chamou a equipe Ginga com Tapioca e outros amigos para jogar.

Nesse jogo, há **N** cidades, e existem **M** estradas ligando elas. Cada estrada liga uma cidade **A** a uma cidade **B** e leva **C** minutos para ser atravessada (note que essa estrada é de **A** para **B**, mas não de **B** para **A**).

Agora cada um dos **K** jogadores está em uma cidade diferente no jogo, e todos vão dirigir o mais rápido possível para uma cidade **X**. Sua tarefa é dizer, para cada jogador, qual será o menor tempo necessário para que ele chegue no destino.

### Entrada:

A primeira linha contém dois inteiros **N** e **M** ( $2 \leq N \leq 2 \times 10^5$  e  $1 \leq M \leq 2 \times 10^5$ ), sendo as quantidades de cidades e de estradas, respectivamente. As cidades são numeradas de 1 a **N**.

A linha seguinte contém um inteiro **F** ( $1 \leq F \leq N$ ), que é a cidade para a qual todos os jogadores irão se deslocar.

Cada uma das **M** linhas seguintes descrevem as estradas, contendo os três inteiros **A B C** ( $1 \leq A, B \leq N$  e  $1 \leq C \leq 10^4$ ), que indicam, respectivamente, a cidade de origem e de destino da estrada, e o tempo gasto para atravessar essa estrada. Isto é, indicam que é possível pegar uma estrada que vai da cidade **A** para a cidade **B** que leva tempo **C**, em minutos.

A linha seguinte contém um inteiro **K** ( $1 \leq K \leq N$ ), que indica a quantidade de amigos jogando.

As **K** linhas seguintes contêm, cada uma, um inteiro **V** ( $1 \leq V \leq N$ ), indicando a cidade em que cada amigo está.

É garantido que não há uma estrada ligando uma cidade a ela mesma.

É garantido que é possível pegar um caminho de qualquer cidade para a cidade **F** (possivelmente passando por outras cidades).

É garantido que para cada par de cidades, há no máximo uma estrada indo da primeira cidade para a segunda (mas é possível que exista uma estrada que ligue as cidades em um sentido, e outra que ligue as mesmas cidades no sentido oposto).

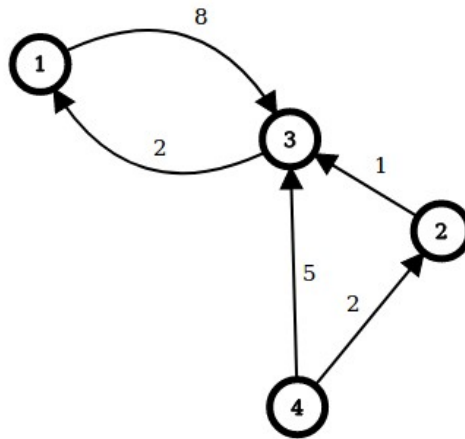
É garantido que cada amigo está em uma cidade diferente (isto é, cada valor de **V** é distinto).

### Saída:

Para cada amigo, na mesma ordem que foram listados, indique o tempo necessário, em minutos, para que ele chegue ao destino (cidade **F**).

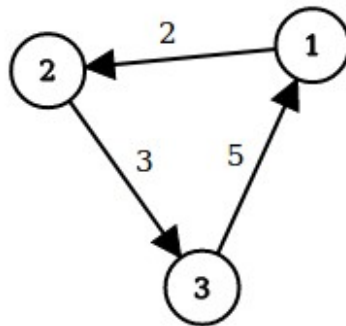
EXEMPLOS	
Exemplo de <b>entrada 1</b> :	
4 5	
3	
3 1 2	
1 3 8	
4 3 5	
4 2 2	
2 3 1	
2	
1	
4	
Exemplo de <b>saída 1</b> :	
8	
3	
Exemplo de <b>entrada 2</b> :	
3 3	
2	
1 2 2	
2 3 3	
3 1 5	
3	
2	
1	
3	
Exemplo de <b>saída 2</b> :	
0	
2	
7	

\*Obs: o desenho a seguir ilustra o primeiro exemplo:



Nesse exemplo há 4 vértices e 5 arestas, e os amigos querem ir para a cidade 3. São dois amigos nesse exemplo. O primeiro amigo está na cidade 1, e precisa pegar a estrada que leva 8 minutos para chegar na cidade 3. Já o segundo amigo está na cidade 4, e pode pegar a estrada direto para a cidade 3, que leva 5 minutos, ou pegar uma estrada para a cidade 2 (que leva 2 minutos) e outra para a cidade 3 (que leva 1 minuto), tendo um tempo total igual a 3 minutos. Como ele vai escolher o caminho mais rápido, vai seguir o segundo caminho.

Para o segundo exemplo, segue a seguinte ilustração:



Nesse exemplo temos 3 cidades e 3 estradas, e os amigos querem ir para a cidade 2. São 3 amigos. O primeiro amigo já está na cidade 2, então precisa de um tempo total igual a 0 para chegar nela. O segundo está na cidade 1, e precisa pegar uma estrada que leva 2 minutos para chegar lá. Por fim, o terceiro amigo está na cidade 3, e para chegar na cidade 2 precisa pegar duas estradas: uma que leva 5 minutos, indo para a cidade 2, e depois outra que leva 2 minutos, indo para a cidade 1, tendo um tempo total igual a 7.

## F - Múltiplos

Para exercitar sua criatividade em seu próximo jogo, Railton decidiu escrever números. Então, ele escreveu um array com  $N$  números. Após esse feito, chamou Hélio para ver o resultado. Ao invés de Hélio se impressionar com a quantidade de números que Railton conseguiu escrever, Hélio começou a observar outra quantidade: o número de pares de elementos desse vetor onde o primeiro é múltiplo do segundo. Mais formalmente, sendo  $A_i$  o  $i$ -ésimo elemento do vetor  $\mathbf{A}$  que Railton escreveu, Hélio quer saber a quantidade de pares  $(i,j)$  tal que  $A_i$  é múltiplo de  $A_j$ , com  $i \neq j$ . Seu dever é ajudá-los a encontrar essa quantidade.

### Entrada:

A primeira linha da entrada contém um inteiro  $N$  ( $1 \leq N \leq 10^6$ ), que representa o tamanho do vetor.

A linha seguinte é composta por  $N$  inteiros que formam o vetor  $\mathbf{A}$ . Cada elemento do vetor  $\mathbf{A}$  está entre 1 e  $10^6$  (inclusive).

### Saída:

A saída deve conter apenas um inteiro que indique a quantidade de pares  $(i,j)$  tal que  $A_i$  é múltiplo de  $A_j$ , com  $i \neq j$ .

EXEMPLOS	
Exemplo de <b>entrada 1</b> :	
3	
1 2 3	
Exemplo de <b>saída 1</b> :	
2	
Exemplo de <b>entrada 2</b> :	
3	
1 2 4	
Exemplo de <b>saída 2</b> :	
3	
Exemplo de <b>entrada 3</b> :	
2	
1 1	
Exemplo de <b>saída 3</b> :	
2	
Exemplo de <b>entrada 4</b> :	
5	
2 6 3 8 43	
Exemplo de <b>saída 4</b> :	
3	

\*Obs: Para o primeiro, temos um vetor com três elementos, sendo  $A_1 = 1$ ,  $A_2 = 2$ ,  $A_3 = 3$ . No total, temos os seguintes pares: (2,1) e (3,1), afinal,  $A_2$  é múltiplo de  $A_1$ , assim como  $A_3$ . Portanto, a resposta é 2.

(Continuação na página seguinte)

Já para o segundo exemplo, temos  $A_1 = 1$ ,  $A_2 = 2$ ,  $A_3 = 4$ . No total, temos os seguintes pares: (2,1), (3,1) e (3,2), portanto, a resposta é 3.



Para o terceiro exemplo, temos os pares  $(1, 2)$  e  $(2, 1)$ .

Finalmente, para o quarto exemplo, temos:  $(2, 1)$ ,  $(2, 3)$  e  $(4, 1)$ .