



Simulado OBI2018

Caderno de Tarefas

Modalidade Programação • Nível 1 • Fase 1

16 de maio de 2018

A PROVA TEM DURAÇÃO DE 2 HORAS

Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 5 páginas (não contando a 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`; 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.

Fibra óptica

Nome do arquivo: *fibra.c, fibra.cpp, fibra.pas, fibra.java, fibra.js ou fibra.py*

Recentemente na cidade de Acitobor, chegou a conexão de internet por fibra óptica. Com essa nova tecnologia, a demanda por internet de alta velocidade aumentou e o mercado de empresas que fornecem internet banda larga com fibra óptica cresceu exponencialmente. Cada uma das casas da cidade (que estão enumeradas de 1 a N) contrata a empresa de sua preferência e quando uma casa for ligada a outra por uma rede de fios, significa que essas contratam a mesma empresa de banda larga. A fim de ter um controle sobre a quantidade dessas empresas na cidade, o prefeito Alebasi pediu a sua ajuda. Sabendo as informações dos cabos que ligam as casas, faça um programa que mostre o número de empresas que oferecem o serviço de internet na cidade.

Entrada

A primeira linha da entrada contém um inteiro N e F, representando respectivamente o número de casas da cidade e o número de cabos ligando duas casa. As F linhas seguintes contém dois inteiros A e B, indicando que existe uma ligação entre as casas A e B.

Saída

Seu programa deve imprimir uma linha contendo um inteiro representando o número de empresas fornecedoras de internet na cidade.

Restrições

- $2 \leq N \leq 10^4$
- $2 \leq F \leq 10^5$

Exemplos

Entrada	Saída
8 6	3
3 5	
2 1	
6 8	
1 5	
4 6	
1 3	

Entrada	Saída
7 3	4
5 3	
6 5	
4 2	

Robô em perigo

Nome do arquivo: *robo.c, robo.cpp, robo.pas, robo.java, robo.js ou robo.py*

Catheus gosta muito de robótica. Ele queria desenvolver um robô que jogasse dados com ele, mas seu professor lhe disse para adicionar também outra função mais interessante. Alguns meses depois, seu robô estava pronto. Ele jogava dados e conseguia viajar no espaço. Certo dia, o robô de Catheus foi capturado por alienígenas no planeta OBI-1705 e o único jeito de se livrar era vencendo um jogo que, para a sorte de Catheus e seu robô, era um jogo de dados. O jogo funciona da seguinte forma: o robô vai jogar com N alienígenas ao mesmo tempo. Cada um pegará uma cartela com uma quantidade M de números. O dado alienígena possui D faces. O dado é jogado uma vez pelo robô e quem possuir o maior número de aparições do número do dado na sua cartela vence o jogo. O robô é liberado caso ele vença o jogo ou haja mais de um jogador com maior número de ocorrências do número na cartela.

Entrada

A primeira linha da entrada consiste de um inteiro N representando o número de oponentes alienígenas. A segunda linha consiste de um inteiro M_0 representando a quantidade de números na cartela do robô. A terceira linha contém os M_0 números na cartela do robô. Os próximos N pares de linha possuem respectivamente M_i representando a quantidade de números na cartela do alienígena i e os números da cartela. A última linha possui um inteiro D representando o número que saiu no dado que o robô jogou.

Saída

Imprima "robo" caso o robô seja liberado. Do contrário, imprima o número do alienígena vencedor (considerando que estão numerados de 1 a N).

Restrições

- $1 \leq N \leq 10^4$
- $1 \leq M_i \leq 2 \times 10^6$
- $1 \leq D$, números das cartelas $\leq 10^9$
- $1 \leq \text{Soma dos } M_i \text{'s} \leq 2 \times 10^6$

Exemplos

Entrada	Saída
1 4 2 2 3 4 2 1 2 2	robo

Entrada	Saída
3 5 7 7 7 100 200 5 40 20 30 40 50 1 37 5 70 70 70 70 70 37	2

Entrada	Saída
3 10 1 2 3 4 5 6 7 8 9 10 1 11 1 11 1 11 11	robo