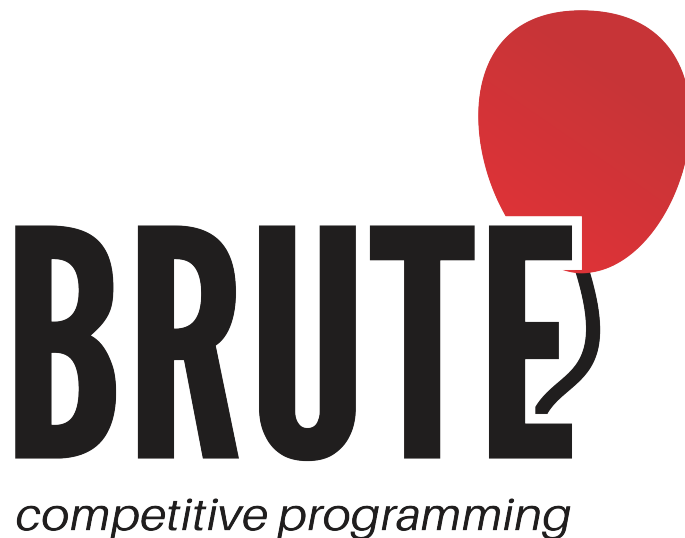


Caderno de Provas
Maratona CODECON
2020

Maratona de Programação - CODECON

Servidor BOCA:
<http://boca.bruteudesc.com/>

Organização e Realização:
BRUTE UDESC



Lembretes:

- Aos *javaneiros*: o nome da classe deve ser o mesmo nome do arquivo a ser submetido.
Ex: classe **petrus**, nome do arquivo **petrus.java**;
- Exemplo de leitura de entradas que funcionam:

```
Java: (import java.util.Scanner)
Scanner in = new Scanner(System.in);
ou
Scanner stdin = new Scanner(new BufferedReader(new InputStreamReader(System.in)));
```

```
C: (#include <stdio.h>)
int integer1; scanf("%d", &integer1);
```

```
C++: (#include <iostream>)
int integer1; std::cin >> integer1;
```

Exemplo de saída de entradas:

```
Java: System.out.format("%d %d\n", integer1, integer2);
C: printf("%d %d\n", integer1, integer2);
C++: std::cout << integer1 << " " << integer2 << std::endl;
```

- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova;
- A correção é automatizada, portanto, **siga atentamente as exigências da tarefa quanto ao formato da entrada e saída conforme as amostras dos exemplos**. Deve-se considerar entradas e saídas padrão;
- Todos os compiladores (Java, Python, C e C++) são padrões da distribuição Ubuntu versão 16.04 (gcc C11 habilitado);
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. As soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Teste seu programa antes de submetê-lo. A cada problema detectado (erro de compilação, erro em tempo de execução, solução incorreta, formatação imprecisa, tempo excedido ...), há penalização de 20 minutos. O tempo é critério de desempate entre duas ou mais equipes com a mesma quantidade de problemas resolvidos;
- Utilize o *clarification* para dúvidas da prova. Os juízes podem **opcionalmente** atendê-lo com respostas acessíveis a todos;
- Algumas interfaces estão disponíveis nas máquinas Linux, que podem ser utilizada no lugar da *Unity*. Para isto, basta dar *logout*, e selecionar a interface desejada. Usuário e senha: *udesc*;
- Ao pessoal local: **cuidado com os pés sobre as mesas para não desligarem nenhum estabilizador/computador de outras equipes!**

Agradecimentos

- Alan Turing, ateu e homossexual pai da ciência da computação;
- George Boole, 1100010 1110010 1100001 1100010 11000011 10101101 1110011 1110011 1101001 1101101 1101111;

Maratona de Programação CODECON - 2020

Maratona de Programação CODECON

CODECON 2020

Conteúdo

1	Problema A: Drone de entrega	4
2	Problema B: Lista de Compras	5
3	Problema C: Loja de pisos do sr. Bigodeira	6
4	Problema D: Troca de Chaves	8
5	Problema E: Grande confusão de Bazollo	9

1 Problema A: Drone de entrega

Arquivo: A.[c|cpp|java|py]

Tempo limite: 1 s

Autor: Ferdinando Augusto Galera Junior

Um drone de entrega sempre sai do armazém da distribuidora, a partir daí ele vai passando por diversas residências fazendo entregas e ao final do expediente o drone retorna ao armazém. Cada residência é indicada por coordenadas (X,Y) e o armazém sempre estará na coordenada $(0,0)$.

Entrada

A entrada contém um inteiro indicando o número R de residências que o drone visitará. As próximas R linhas dão 2 valores inteiros X e Y indicando a coordenada de cada residência.

Restrições

- $1 \leq R \leq 10^6$.
- $-1000 \leq X, Y \leq 1000$.

Saída

Você deve imprimir a distância total percorrida pelo drone como duas casas decimais de precisão. Lembrando que ele começa e termina no armazém, além disso ele visita as casas na ordem que as coordenadas são dadas.

Exemplo de Entrada	Exemplo de Saída
5 1 3 4 6 5 7 8 3 9 4	25.08
3 98 45 75 68 24 83	279.93

2 Problema B: Lista de Compras

Arquivo: B.[c|cpp|java|py]

Tempo limite: 1 s

Autora: Laís Van Vossen

Problema

Bigodinho, filho do sr. Bigodeira foi encarregado de ir no mercado comprar os itens que precisa para cozinhar com seu pai. Conhecendo seu pai, Bigodinho sabe que seu sr. Bigodeira é bem preciso em suas medições e sempre coloca exatamente a quantidade necessária que o garoto precisa comprar.

Porém chegando no mercado, o menino percebe que nem sempre os produtos são vendidos na quantidade exata que seu pai precisa. Como é sempre melhor sobrar do que faltar, ele pede sua ajuda para desenvolver um programa que dado a lista de compras fornecida por Bigodeira, indique para Bigodinho quantas unidades de cada item ele precisa levar para casa. Lembre-se, é sempre melhor sobrar do que faltar e não é possível comprar uma fração de um produto.

Entrada

A entrada consiste de um inteiro C indicando quantos itens tem na lista. Nas C linhas seguintes haverão N e M inteiros positivos indicando a quantidade, em gramas, solicitada por bigodeira e a quantidade de produto, em gramas, que uma unidade possui.

Restrições

- $1 \leq N, M \leq 1000$.
- $1 \leq C \leq 100$.

Saída

Para cada item na lista, você deve imprimir quantas unidades de cada produto que Bigodinho deverá comprar.

Exemplo de Entrada	Exemplo de Saída
1 150 100	2
2 20 100 200 200	1 1

3 Problema C: Loja de pisos do sr. Bigodeira

Arquivo: C.[c|cpp|java|py]

Tempo limite: 1 s

Autora: Laís Van Vossen

Problema

Bigodeira é dono de uma loja de pisos muito peculiar, sua loja vende apenas pisos de tamanho 2×1 , que podem ser rotacionados para 1×2 . Com esse formato de piso ele é capaz de preencher salas de tamanhos diversos, porém por ser muito perfeccionista ele prefere se ater a sala de tamanho $4 \times N$. Como toda pequena loja, ele conta com um estoque limitado, o tamanho máximo do ambiente que ele pode atender é apenas 4×30 .

Levando isso em conta, Bigodeira pediu a sua ajuda para montar um programa que calcule de quantas formas possíveis ele pode preencher o piso da sala de seu cliente. Ou seja, dado um N qualquer, você deve determinar de quantas formas a sala pode ser completamente coberta pelo piso do formato que Bigodeira possui.

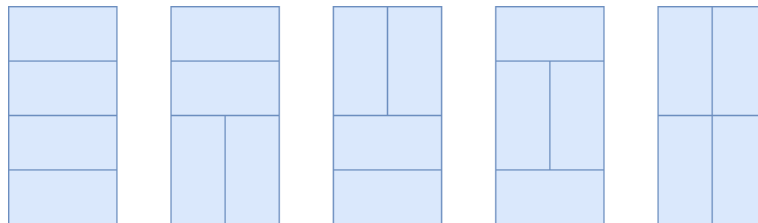


Figura 1: Exemplo para uma sala de tamanho 4×2

Entrada

A entrada consiste de um inteiro C indicando quantos clientes deverão ser atendidos por Bigodeira. Nas C linhas seguintes haverá N inteiros positivos indicando o comprimento da sala.

Restrições

- $1 \leq N \leq 30$.
- $1 \leq C \leq 1000$.

Saída

Para cada cliente, você deve imprimir a quantidade de formas de cobrir a sala de tamanho $4 \times N$.

Exemplo de Entrada	Exemplo de Saída
3 2 3 7	5 11 781
2 6 9	281 6336

4 Problema D: Troca de Chaves

Arquivo: D.[c|cpp|java|py]

Tempo limite: 1 s

Autor: Rafael Granza de Mello

O protocolo de troca de chaves Diffie–Hellman é muito importante para os sistemas de criptografia atuais. O intuito dele é fazer com que duas pessoas possam trocar uma chave criptográfica sigilosamente mesmo se uma terceira pessoa estiver ouvindo a conversa.

O protocolo consiste de duas chaves abertas, B e M , onde B representa uma base e M representa um módulo. Além disso cada pessoa deve escolher aleatoriamente um expoente E (devem ser mantidos em segredo). A partir daí deve se seguir as etapas:

- Cada pessoa deve computar $(B^E \bmod M)$ de acordo com o expoente que escolheu, vamos chamar esse resultado de R ;
- Cada pessoa passa seu resultado R para a outra;
- Cada pessoa deve computar $(R^E \bmod M)$ de acordo com o expoente E que havia escolhido e o resultado R que recebeu do outro usuário, vamos chamar esse valor de K ;
- K computado deve ser igual para ambas as pessoas, e essa será a chave que será usada para criptografar as mensagens futuras;

Entrada

A entrada consiste de dois primos B e M indicando a base e o módulo na primeira linha, e mais dois inteiros E_1 e E_2 indicando os expoentes escolhidos por cada usuário.

Restrições

- $2 \leq B, M \leq 10^9 + 9$.
- $2 \leq E_1, E_2 \leq 10^{18}$.

Saída

Você deve imprimir a chave que será usada pelos usuários.

Exemplo de Entrada	Exemplo de Saída
97 1000000009 7919 761	992325537
761 10000000007 65472 500	943985070

5 Problema E: Grande confusão de Bazollo

Arquivo: E.[c|cpp|java|py]

Tempo limite: 1 s

Autor: Vinicius Gasparini

Bazollo é o mais novo funcionário de T.I. da grande empresa financeira DINDIN (DINheiro na DINamarca). Tudo estava correndo uma maravilha, seus novos colegas de trabalho eram muito competentes em realizar transferências de dinheiro aqui do Brasil para a Dinamarca sem que nada desse errado (ou pelo menos sem ninguém descobrir). Por se tratar de um negócio muito sigiloso e de risco, a rede de internet era cheia de trancas para acessos a sites de entretenimento.

Mas Bazollo, *gamer* desde seus 4 anos, não estava contente em ter que voltar do horário de almoço e ficar sem acesso ao seu joguinho predileto, o CS (Computer Science, the game). Então certo dia Bazollo instalou um arquivo que seu amigo Josnk forneceu para que criasse um *backdoor* no *firewall* da empresa.

Como você já pode imaginar, deu ruim! Bazollo acabou por infectar sua máquina com um vírus cruel (mas nem tanto). Com medo de sofrer uma demissão precoce, Bazollo decidiu isolar sua máquina da rede e só então ligá-la novamente ao consertar seu erro.

Para isso, ele precisa da sua ajuda para que sob posse do mapa da rede você isole o computador infectado dos demais de maneira a minimizar a dificuldade da tarefa e evitar que o vírus se espalhe até chegar no servidor. No diagrama abaixo é possível entender melhor como isso funciona.

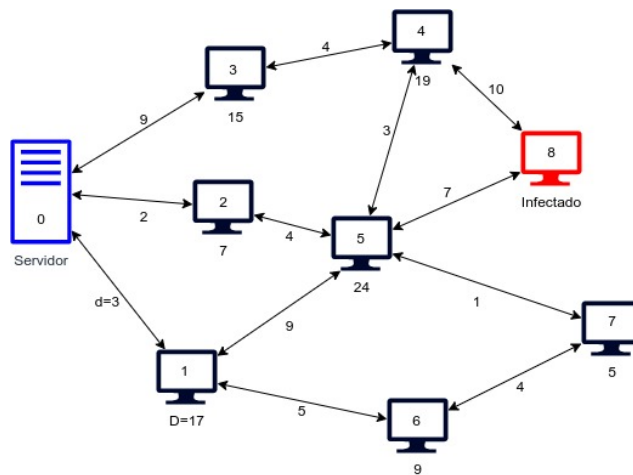


Figura 2: Rede da empresa DINDIN

Cada computador possui um nível de dificuldade para ser reparado assim como cada conexão possui uma dificuldade associada. Como Bazollo não quer incomodar ninguém, fechar as conexões é o ideal.

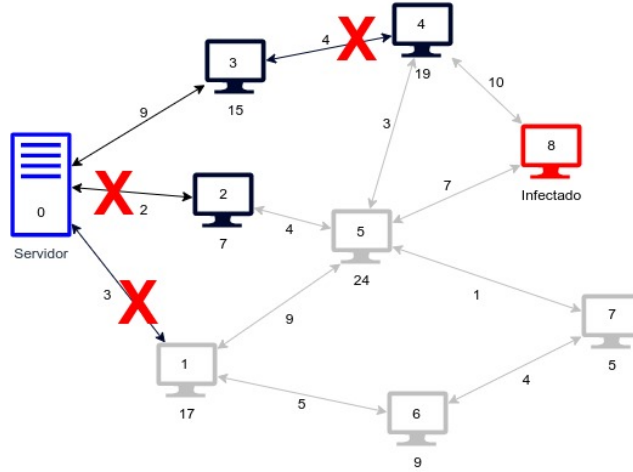


Figura 3: Solução para isolar do computador de Bazollo

Problema

Sua tarefa é, com base nas informações da rede, impedir que o computador infectado de Bazollo alcance o servidor da empresa. Essa tarefa deve ser feita de maneira a minimizar a dificuldade acumulada.

Entrada

A entrada consiste de dois inteiros N, B . Sendo N quantos dispositivos a rede possui, incluso o computador de Bazollo e o servidor. O dispositivo 0 é o servidor e B o identificador do computador do Bazollo.

Para cada um dos N dispositivos, a entrada iniciará de dois inteiros D, M , ou seja, o i -ésimo dispositivo possui dificuldade D e tem M conexões. As próximas M linhas serão compostas de um par v, d , isto é, o i -ésimo dispositivo faz conexão com o dispositivo v com dificuldade d de cortar essa relação. Note que as conexões não possuem direção. Tanto para o servidor quanto para o computador de Bazollo, o valor de D é 0, ou seja, você não poderá encostar neles.

Restrições

- $2 \leq N \leq 100$.
- $1 \leq B \leq N - 1$.
- $0 \leq D, d \leq 100$.
- $\sum_{i=0}^{N-1} M_i \leq 10^4$.
- $0 \leq v \leq N - 1 \mid v \neq i$.
- Sendo $G = (V, E)$ o grafo da rede, e o conjunto de arestas adjacentes ao vértice v e X o peso do vértice, temos que:

$$\sum_{x \in e} pesos(x) \leq X$$

Saída

Você deve imprimir a dificuldade mínima acumulada para isolar o computador do pobre Bazollo.

Exemplo de Entrada	Exemplo de Saída
9 8 0 3 1 3 2 2 3 9 17 3 0 3 5 9 6 5 7 2 0 2 5 4 15 2 0 9 4 4 19 3 3 4 5 3 8 10 24 5 1 9 2 4 4 3 7 1 8 7 9 2 1 5 7 4 5 2 5 1 6 4 0 2 4 10 5 7	9