

CI097 - Prova 3 B

25 de Janeiro de 2017

Servidor BOCA:
<http://maratona.c3sl.ufpr.br/>

Instruções Importantes

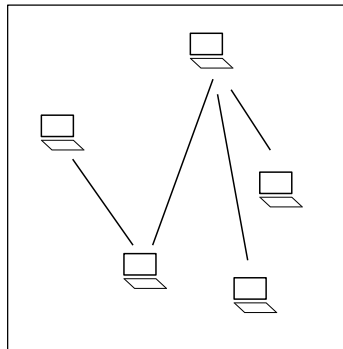
- A prova é composta por 5 problemas que valem 20 pontos cada, e um problema bônus que vale 20 pontos.
- O problema F é um problema bônus. Este problema só deve ser considerado caso todos os demais forem resolvidos.
- Se sua solução para um problema for aceita, a nota integral do problema é garantida.
- Se sua solução não for aceita, o professor irá avaliar sua última (mas **apenas** a última) submissão para o problema e atribuir a ela uma nota parcial, considerando a corretude do algoritmo utilizado, a complexidade computacional do algoritmo, o uso correto de funções e estruturas de dados disponíveis utilizadas, e eventuais detalhes da solução.
- Em cada problema, cada arquivo de entrada contém *apenas um* caso de teste. Sua solução será executada com vários arquivos de entrada separadamente.
- Sua solução será compilada ou executada com a seguinte linha de comando:
 - C: `gcc -static -O2 -lm`
 - C++: `g++ -static -O2 -lm`
 - C++11: `g++ -std=c++11 -static -O2 -lm`
 - Java: `javac`
 - Pascal: `fpc -Xt -XS -O2`
 - Python: `python3`
- O tempo limite para a execução de sua solução é de 1 segundo para todos os problemas da prova exceto o problema F, cujo tempo limite é de 3 segundos.
- Todas as linhas, tanto na entrada quanto na saída, terminam com o caractere de fim-de-linha (`\n`), mesmo quando houver apenas uma única linha no arquivo.
- Para submissões em **JAVA**, a classe deverá ter o mesmo nome que o *basename* do problema (leia a linha entre o título e o texto do problema).

A: Rede do DINF

Arquivo: `rede.[c|cpp|java|pas|py]`

O prédio do Departamento de Ilhas Naturais Federais (DINF) está passando por uma grande reforma. Por isso, os antigos laboratórios de informática foram destruídos, e um novo será construído no segundo andar do prédio.

Existem N computadores no novo laboratório, numerados de 1 a N . Para cada computador i , $1 \leq i \leq N$, é conhecida a posição (x_i, y_i) na qual ele está instalado. Agora, é necessário construir a rede que irá conectar todos os computadores do laboratório. Para tal, é possível construir cabos de rede entre pares de computadores. Para que a rede seja válida, os cabos devem ligar os computadores de tal forma que seja possível enviar uma mensagem de qualquer máquina para qualquer outra no laboratório, passando por um ou mais cabos instalados. A figura abaixo exemplifica uma possível configuração de uma rede válida:



Sua tarefa é, dadas as posições de cada computador no laboratório, determinar o comprimento total mínimo de cabo de rede necessário para construir uma rede válida.

Entrada

A primeira linha contém o inteiro N ($1 \leq N \leq 500$), o número de computadores no laboratório. As próximas N linhas contém dois inteiros x_i e y_i cada ($1 \leq x_i, y_i \leq 10^4$), indicando a posição de um computador. Não há mais de um computador na mesma posição.

Saída

Imprima uma única linha contendo o comprimento total mínimo de cabo de rede necessário. Arredonde e imprima a resposta com exatamente duas casas decimais.

Exemplo de entrada	Exemplo de saída
4 1 1 5 1 1 5 5 5	12.00

B: Troca ou não troca

Arquivo: troca.[c|cpp|java|pas|py]

Você provavelmente conhece programas de TV do tipo *Troca ou não Troca?*, onde um participante pode ganhar um prêmio que pode ir desde uma caneta até uma casa com piscina.

Existem N prêmios, numerados de 1 a N . O participante começa o programa com o prêmio 1. O programa tem então $N - 1$ rodadas. Na rodada i , o apresentador pergunta ao participante se ele deseja trocar seu prêmio atual pelo prêmio P_i . O participante, que não sabe qual prêmio está sendo oferecido, deve responder “sim” ou “não”. Se o participante responder “sim”, seu prêmio atual é trocado pelo prêmio P_i . Se o participante responder “não”, seu prêmio atual não é alterado. Após a última rodada, o participante leva para casa o prêmio que tiver.

Seu irmão será o participante do programa que está prestes a começar! Antes do programa, ele lhe contou a resposta que pretende dar em cada rodada. Entretanto, você tem o poder de controlar sua mente! Ao utilizar o poder em alguma rodada, a resposta dele para aquela rodada é alterada, isto é, ele irá dizer “sim” caso pretendia dizer “não”, e vice-versa.

Você acabou de descobrir qual prêmio será oferecido em cada rodada. Você quer muito que ele termine o programa com o prêmio N , o de maior valor entre todos os prêmios. Qual a quantidade mínima de vezes que seu poder deve ser utilizado para garantir que seu irmão irá levar o prêmio N ?

Entrada

A primeira linha contém o inteiro N ($2 \leq N \leq 100$), o número de prêmios. As próximas $N - 1$ linhas descrevem uma rodada cada, em ordem. Cada linha contém dois inteiros P_i e R_i ($2 \leq P_i \leq N$, $R_i = 0$ ou 1), o prêmio oferecido na rodada e a resposta que seu irmão pretende dar na rodada, respectivamente. $R_i = 0$ indica “não”, enquanto $R_i = 1$ indica “sim”.

Nenhum prêmio será oferecido mais de uma vez, e o prêmio 1 não será oferecido novamente.

Saída

Imprima uma linha com a quantidade mínima de vezes que você deve usar seu poder para garantir que o prêmio N será conquistado.

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

Exemplo de entrada	Exemplo de saída
4 3 0 4 0 2 1	2

C: Giga Drive

Arquivo: giga.[c|cpp|java|pas|py]

A *TéquiToy* está relançando no mercado o *Giga Drive*, um console antigo muito popular nos anos 90. Para comemorar o relançamento do console, a *TéquiToy* vai lançar um jogo “inédito” este ano: o *Giga Drive CheeseBread 2*! Na verdade, o *Giga Drive CheeseBread 2* é um clone com poucas alterações de um outro jogo famoso¹. Por isso, suas regras são muito semelhantes.

Em *Giga Drive CheeseBread 2*, o tabuleiro do jogo consiste em uma matriz de N linhas e N colunas. Inicialmente, cada célula da matriz está vazia ou contém uma azeitona (?). Uma partida é composta por M rodadas. A cada rodada, o computador coloca uma azeitona em uma célula vazia do tabuleiro. Em seguida, ele pergunta ao jogador o total de azeitonas na região do tabuleiro da célula (I_1, J_1) até uma célula (I_2, J_2) dada. Em outras palavras, o computador indica quatro inteiros I_1, J_1, I_2 e J_2 , e pergunta ao jogador o número total de azeitonas nas células na linha i e coluna j tal que $I_1 \leq i \leq I_2$ e $J_1 \leq j \leq J_2$.

Para vencer o jogo, você deve acertar a contagem em todas as rodadas do jogo! Entretanto, você tem muito pouco tempo para contar todas as azeitonas. Por isso, escreva um programa para garantir sua vitória!

Entrada

A primeira linha contém os inteiros N e M ($1 \leq N \leq 1000, 1 \leq M \leq 10^5$). As próximas N linhas contém N inteiros cada, descrevendo o tabuleiro inicial. O valor 1 indica uma azeitona, enquanto o valor 0 indica uma célula vazia. As próximas $2 \times M$ linhas descrevem as rodadas do jogo, uma rodada a cada duas linhas. A primeira linha contém dois inteiros i e j ($1 \leq i, j \leq N$) indicando a inclusão de uma azeitona na célula na linha i e coluna j . É garantido que a célula estava vazia antes da rodada. A segunda linha de cada rodada contém quatro inteiros I_1, J_1, I_2 e J_2 ($1 \leq I_1 \leq I_2 \leq N, 1 \leq J_1 \leq J_2 \leq N$), descrevendo a região do tabuleiro da célula (I_1, J_1) até a célula (I_2, J_2) .

Saída

Para cada rodada, imprima uma linha indicando o total de azeitonas na região.

Exemplo de entrada	Exemplo de saída
4 2 0 0 1 1 0 1 0 1 0 0 1 0 1 1 0 1 1 1 1 1 2 2 2 3 1 1 2 4	2 6

¹nos anos 90, era comum a *TéquiToy* clonar outros jogos. *Turma da Mônica em: O Resgate*, por exemplo, é um clone de *Wonder Boy III*.

D: Tubos de ensaio

Arquivo: `tubos.[c|cpp|java|pas|py]`

O prédio da Engenharia Química (EQ) é um grande prédio recém inaugurado no campus. Ao entrar no prédio à procura de um banheiro, você acabou entrando por engano em um laboratório de química! Felizmente, o instrutor do laboratório permitiu sua entrada nele.

Neste laboratório, há N tubos de ensaio, numerados de 1 a N . O tubo i ($1 \leq i \leq N$) contém inicialmente m_i mililitros de água. Para cada tubo, você pode deixá-lo inalterado, retirar qualquer quantidade de água dele, ou colocar qualquer quantidade de água nele. Seu objetivo é deixar todos os tubos com a mesma quantidade de água.

Para colocar uma quantidade de X ml em algum tubo, você precisa retirar X ml de água de uma torneira no laboratório. Além disso, para retirar uma quantidade de X ml de algum tubo, você precisa despejar X ml de água em um ralo no laboratório. O *custo total* é igual à soma da quantidade total de água retirada da torneira com a quantidade total de água despejada no ralo.

Dadas as quantidades iniciais de água em cada tubo, sua tarefa é determinar o custo total mínimo necessário para deixar todos os tubos com uma mesma quantidade de água.

Entrada

A primeira linha contém o inteiro N ($1 \leq N \leq 100$), a quantidade de tubos no laboratório. A segunda linha contém N números reais m_i ($0 \leq m_i \leq 100.0$), a quantidade inicial de água em cada tubo, em mililitros.

Saída

Imprima uma única linha contendo o custo total mínimo para deixar todos os tubos com a mesma quantidade de água. Arredonde e imprima o valor com exatamente duas casas decimais.

Exemplo de entrada	Exemplo de saída
3 1.0 1.0 0.9	0.10

Exemplo de entrada	Exemplo de saída
3 42.0 42.4 41.7	0.70

E: A mensagem de Will

Arquivo: mensagem.[c|cpp|java|pas|py]

Ao voltar de um intenso jogo de RPG na casa de um amigo, o jovem Will desapareceu misteriosamente! Todos estão desesperadamente procurando por ele por todos os cantos. Enquanto isso, coisas estranhas estão acontecendo em sua casa. Uma delas, entretanto, lhe permite comunicar-se com o garoto!

Há exatamente 26 lâmpadas penduradas na parede da sua sala, numeradas de 1 a 26 da esquerda para a direita. Além disso há uma letra do alfabeto pintada na parede em baixo de cada lâmpada. Quando Will quer lhe enviar uma mensagem, ele irá (misteriosamente) piscar, uma a uma, as lâmpadas correspondentes a cada letra de sua mensagem. Por exemplo, se ele quer enviar a mensagem **HELP**, ele irá piscar, nesta ordem, as lâmpadas acima das letras H, E, L e P.

Dada a letra associada a cada lâmpada e a ordem das lâmpadas que foram piscadas por Will, decifre a mensagem que ele enviou!

Entrada

A primeira linha contém uma *string* de exatamente 26 letras maiúsculas contendo todas as letras do alfabeto inglês. A primeira letra da *string* está associada à lâmpada 1; a segunda letra está associada à lâmpada 2; e assim por diante. A próxima linha contém um inteiro N ($1 \leq N \leq 10^4$), o número de lâmpadas que foram piscadas. A terceira linha contém N inteiros l_i ($1 \leq l_i \leq 26$), indicando as lâmpadas que foram piscadas, em ordem.

Saída

Imprima uma única linha contendo a mensagem enviada por Will.

Exemplo de entrada	Exemplo de saída
ABCDEFGHIJKLMNOPQRSTUVWXYZ 4 8 5 12 16	HELP

Exemplo de entrada	Exemplo de saída
QWERTYUIOPASDFGHJKLZXCVBNM 10 16 3 19 19 9 2 9 4 19 13	HELLOWORLD

F: Mesada do Netinho

Arquivo: mesada.[c|cpp|java|pas]

Está cada vez mais difícil para a sra. Marie gerenciar a mesada que ela dá para seus netinhos. Por isso, ela lhe pediu para escrever um programa de computador para ajudá-la nesta tarefa.

A sra. Marie tem N netos, numerados de 1 a N . Inicialmente, o neto i ganha M_i reais de mesada de sua vovó. A sra. Marie pode aumentar a mesada de alguns de seus netos várias vezes. Sempre que deseja dar um aumento, ela escolhe dois números i e j e aumenta a mesada de *todos* os netos de números entre i e j , inclusive, em um mesmo valor v . Por exemplo, se ela escolher dar um aumento de $v = 10$ reais para todos os netos entre $i = 1$ e $j = 3$, então a mesada dos netos 1, 2, e 3 aumentam em 10 reais *cada uma*.

Além disso, ela também pode querer consultar quem é o neto que recebe a *maior* mesada dentre todos os netos de números entre i e j dados, inclusive. Ajude a sra. Marie a responder todas as suas consultas!

Entrada

A primeira linha contém dois inteiros N e Q ($1 \leq N \leq 10^5$, $1 \leq Q \leq 10^6$), o número de netos e de operações, respectivamente. A próxima linha contém N inteiros M_1, M_2, \dots, M_N ($1 \leq M_i \leq 200$), a mesada inicial de cada neto. As próximas Q linhas descrevem uma operação cada. Uma operação no formato **A** i j v ($1 \leq i \leq j \leq N$, $1 \leq v \leq 200$) indica um aumento de mesada. Uma operação no formato **C** i j ($1 \leq i \leq j \leq N$) indica uma consulta.

Saída

Para cada consulta, imprima uma linha contendo o número do neto que recebe a maior mesada dentre os netos de números entre i e j , inclusive. Caso haja mais de um neto recebendo a maior mesada dentre estes, imprima o de menor número.

Exemplo de entrada	Exemplo de saída
5 6	2
40 55 45 55 70	5
C 2 4	3
C 1 5	4
A 1 3 10	
C 3 4	
A 4 4 20	
C 1 5	