

IV Maratona de Programação do IFB

Caderno de Problemas

27 de abril de 2019



(Este caderno contém 15 problemas)

Comissão Organizadora:

Prof. Daniel Saad Nogueira Nunes (IFB)
Prof. Edson Alves da Costa Júnior (UnB/FGA)
Prof. Guilherme Novaes Ramos (UnB)
Prof. Vinicius Ruela Pereira Borges (UnB)
Arthur Luís Komatsu Aroeira
Jeremias Moreira Gomes (UnB)
Lucas Vasconcelos Mattioli

Apoio:

Ramo Estudantil IEEE/IFB



QNM 40, Área Especial nº 01, Taguatinga/DF, 72146-000 , Brasil
Telefone (61) 2103-2200
<http://www.ifb.edu.br/taguatinga>

Problema A

Primos Arrojados

Limite de tempo: 1s

Autor: Arthur Luís Komatsu Aroeira

Kanela é um garoto fascinado por números primos (números que contêm exatamente 2 divisores positivos distintos). Desde que aprendeu na escola, sempre tentou encontrar primos em todos os números que via. Um de seus hobbies favoritos era escrever números primos em seu caderno. Um belo dia, Kanela resolveu apagar um número primo em seu caderno com uma borracha da direita para esquerda. Ao apagar o último dígito, Kanela notou que o número restante continuava sendo um primo. Ao apagar o último dígito deste, Kanela viu que o número restante também era primo. E assim por diante, conforme os dígitos eram apagados da direita para a esquerda, o número remanescente era primo até que fosse apagado completamente. Kanela achou isso o máximo e começou a chamar esses números de *primos arrojados*!

Por exemplo, o número 2393 é um primo arrojado, pois:

- 2393 é um número primo;
- retirando-se o último algarismo, 239 continua sendo um número primo;
- retirando-se o último algarismo, 23 também é um número primo; e
- removendo-se o último algarismo, tem-se 2 que é primo.

Ajude Kanela a dizer se um número é um *primo arrojado* ou não.

Entrada

A primeira linha contém o inteiro T ($1 \leq T \leq 2 \cdot 10^5$), que representa a quantidade de casos de teste.

Cada uma das próximas T linhas contém um número n ($1 \leq n \leq 10^7$).

Saída

Para cada caso de teste, imprima uma linha com 'S', caso o número n seja um *primo arrojado*, ou 'N', caso contrário.

Exemplo

Entrada	Saída
11	N
1	S
2	S
7	N
10	N
11	N
22	S
23	N
123	N
173	S
233	S
2393	

Notas

Observe que:

- 1 não é um primo arrojado, pois não é um número primo
- 2 é um primo arrojado, pois é primo e contém apenas um dígito
- 23 é um primo arrojado, pois 23 é primo e, retirando-se o último dígito, 2 também é primo
- 173 não é um primo arrojado, pois 1 não é primo, apesar de 17 e 173 serem primos
- 2393 é um primo arrojado, pois 2393, 239, 23 e 2 são todos números primos.

Atenção: É preciso usar um mecanismo de leitura e escrita eficiente para evitar o TLE na hora de escrever a saída do problema.

Para o C++, duas alternativas são:

1. Usar as funções `printf` / `scanf`.
2. Adicionar a linha `ios::sync_with_stdio(false)` no início da função `main`.

No caso de Java, é preciso fazer leitura com buffer (`Scanner` resulta em TLE).

Problema B

Listas Ordenadas

Limite de tempo: 1s

Autor: Daniel Saad Nogueira Nunes

Uma lista de números cujo comprimento é maior do que um está ordenada caso seus elementos estejam dispostos em ordem decrescente ou crescente. Por exemplo a lista $(5, 4, 3, 2, 1)$ está em ordem decrescente, a lista $(1, 2, 3, 4, 5)$ está em ordem crescente e a lista $(1, 2, 2, 3, 1)$ não está ordenada.

Essa definição pode ser estendida para sublistas. Dados índices válidos i e j de uma lista $L = (a_0, a_1, \dots, a_{n-1})$, a sublista $L[i, j] = (a_i, \dots, a_j)$ é dita decrescente se $j > i$ e todo $a_k > a_{k-1}$, $i < k \leq j$; ou dita crescente se $j > i$ e todo $a_k < a_{k-1}$, $i < k \leq j$; em outros casos, a sublista $L[i, j]$ não possui uma ordenação.

Construa um programa que, dada uma lista L e diversos intervalos $[i, j]$ determine se $L[i, j]$ está em ordem crescente, decrescente ou se não possui uma ordenação.

Entrada

A primeira linha da entrada contém um único inteiro N ($1 \leq N \leq 10^5$), que indica o número de elementos da lista L .

A próxima linha possui N inteiros a_i ($-10^9 \leq a_i \leq 10^9$, $0 \leq i \leq N - 1$), separados por um espaço em branco, que descrevem cada elemento de L .

Em seguida, há uma linha contendo um número de consultas Q ($1 \leq Q \leq 10^5$).

As Q linhas seguintes possuem, cada uma, um par de índices i e j ($0 \leq i < j \leq N - 1$), separados por espaço, que representam uma consulta sobre o intervalo $[i, j]$ de L .

Saída

Para cada uma das Q consultas, o seu programa deverá imprimir

- “crescente”: caso $L[i, j]$ esteja em ordem crescente.
- “decrescente”: caso $L[i, j]$ esteja em ordem decrescente.
- “nenhum”: caso $L[i, j]$ não possua ordenação.

Exemplo

Entrada	Saída
5	crescente
1 2 3 4 5	crescente
3	crescente
0 1	
0 3	
2 4	
5	crescente
1 2 3 2 1	decrecente
3	nenhum
0 2	
2 4	
0 4	
4	crescente
1 2 2 1	nenhum
4	decrecente
0 1	nenhum
1 2	
2 3	
1 3	

Notas

Considerando o segundo caso de teste, $L = (1, 2, 3, 2, 1)$ e há 3 consultas.

- A resposta para a primeira consulta, $[0, 2]$, é “crescente”, pois $L[0, 2] = (1, 2, 3)$ está na ordem crescente.
- A resposta para a segunda consulta, $[2, 4]$, é “decrecente”, pois $L[2, 4] = (3, 2, 1)$ está na ordem decrescente.
- A resposta para a última consulta, $[0, 4]$, é “nenhum”, pois $L[0, 4] = L$ não está ordenada.

Problema C

Quebra-Cabeça

Limite de tempo: 1s

Autor: Edson Alves da Costa Júnior

Pedrinho guarda, há muitos anos, o primeiro jogo didático que ganhou de presente: um quebra-cabeça composto por várias peças quadradas, e em cada peça está gravada uma letra maiúscula do alfabeto.

As regras do jogo são muito simples: o jogador deve inserir todas as peças em um pote, escolher aleatoriamente N peças e, usando todas as peças escolhidas, formar um palíndromo. Palíndromos são strings cuja leitura tanto da esquerda para a direita quanto da esquerda para direita resultam no mesmo texto. Por exemplo, ABA, MUSSUM e MIRIM são palíndromos; ASAS, DUDU e UM não são palíndromos.

Algumas vezes Pedrinho consegue vencer o desafio, nas demais oportunidades passa horas sem encontrar uma solução. Auxilie o garoto determinando, a partir das peças selecionadas, se é possível ou não formar um palíndromo.

Entrada

A entrada consiste em uma string S composta por, no máximo, 10^5 letras maiúsculas, onde cada letra representa uma peça escolhida por Pedrinho.

Saída

Imprima, em uma linha, o texto "Sim", se for possível formar um palíndromo com todas as N peças, ou "Nao", caso contrário.

Exemplo

Entrada	Saída
ABA	Sim
SUSMUM	Sim
ABC	Nao

Notas

No primeiro caso, a palavra ABA já é um palíndromo.

No segundo caso, as letras podem ser rearranjadas para formar o palíndromo MUS-SUM.

No terceiro caso, as letras podem ser rearranjadas de 6 maneiras: ABC, ACB, BAC, BCA, CAB e CBA. Nenhuma delas é um palíndromo.

Problema D

Sequência Leia e Fale

Limite de tempo: 1s

Autor: Daniel Saad Nogueira Nunes

John Conway é um professor emérito da Universidade de Princeton notado por suas diversas contribuições em várias áreas da Matemática, a mãe de todas as ciências, de acordo com Gauss.

Conway especialmente é reconhecido por seus vários trabalhos na Matemática Recreativa, que visa mais a diversão de resolver quebra-cabeças e desafios do que atacar um problema de pesquisa.

Um dos trabalhos de Conway está relacionado à sequência "leia-e-fale". O número inicial desta sequência é o 1. Qualquer outro número desta sequência é obtido ao falar o número anterior, agrupando os dígitos consecutivos iguais e indicando a quantidade de vezes que esses dígitos aparecem.

Por exemplo, o número 111221 pode ser pronunciado como "três um, dois dois e um um", gerando o próximo número 312211.

Faça um programa que determine o N -ésimo número da sequência leia-e-fale.

Entrada

A entrada consiste de uma única linha contendo um inteiro N ($1 \leq N \leq 30$) que indica o N -ésimo número da sequência leia-e-fale a ser obtido.

Saída

Imprima, em uma linha, o N -ésimo número da sequência leia-e-fale.

Exemplo

Entrada	Saída
1	1
3	21
6	312211

Notas

O primeiro caso de teste refere-se ao primeiro número da sequência leia-e-fale, que é "1".

O segundo caso de teste, refere-se ao terceiro número da sequência da sequência: "21".

Finalmente, o terceiro caso de teste corresponde ao sexto número da sequência: "312211".

Problema E

Self-Service

Limite de tempo: 1s

Autor: Edson Alves da Costa Júnior

João almoçou em um restaurante *self-service*. A balança mostrou que o seu prato continha G gramas de comida, e ele não consumiu nem bebidas nem sobremesa. No momento do pagamento, ele recebeu um desconto de D por cento no valor a ser pago, de modo que pagou R reais ao caixa antes de ir embora, satisfeito com o serviço e com o estabelecimento.

Os amigos de João se interessaram em ir conhecer o restaurante, e perguntaram a João quanto o estabelecimento cobra por quilo, e ele não se lembrava deste valor. Auxilie João, determinando o valor V que o restaurante cobra por quilo de comida (sem o desconto).

Entrada

A entrada consiste em uma única linha, contendo os valores de G, D e R ($1 \leq G \leq 3000, 0 \leq D \leq 30, 0.01 \leq R \leq 200.00$), separados por um espaço em branco. Considere que G e D são inteiros e que R é apresentado com exatamente duas casas decimais.

Saída

Imprima, em uma linha, o valor V que o restaurante cobra por quilo de comida. Se o x é o valor da conta a ser paga com o valor de V informado, e y o valor da conta, sua resposta será considerada correta se $\frac{|x-y|}{\max(1,y)} \leq 10^{-5}$.

Exemplo

Entrada	Saída
500 0 5.00	10.00000000
450 10 8.10	20.00000000
580 12 20.41	39.98824451

Notas

No primeiro caso, o restaurante cobra 10 reais por quilo. Como João comeu 500 gramas (meio quilo) e não teve desconto, o valor que ele pagou foi de $10/2 = 5$ reais.

No segundo caso, o restaurante cobra 20 reais por quilo, o que dá 9 reais pelos 450 gramas que João comeu. O desconto de 10% corresponde a 90 centavos, logo a conta foi paga no valor de 8 reais e 10 centavos.

Os demais casos são análogos.

Problema F

Canibais

Limite de tempo: 1s

Autor: Vinicius Ruela Pereira Borges

A trilha *Nunca-Mais* é bastante popular na cidade de Estranhópolis, possuindo diversos atrativos em seu percurso, como cavernas, cachoeiras, riachos etc. Um grupo de canibais que peregrinam entre cidades percebeu o trânsito de pessoas na trilha e resolveu se aproveitar da situação.

Um andarilho solitário seria um lanche fácil para os famintos antropófagos, mas fornece poucos nutrientes para o grupo todo. Uma excursão seria um banquete, mas o esforço para lidar com muitas pessoas pode ser demais para os antropófagos debilitados pela fome. Além disso, eles sabem que, ao capturarem e devorarem andarilhos, a cidade ficará em alerta e as pessoas deixarão de percorrer a trilha *Nunca-Mais*, forçando-os a retomar a peregrinação. Por questões de sobrevivência, a ideia dos canibais é simplesmente obter o máximo de energia, capturando as pessoas que oferecem menos resistência, para então preparar o melhor banquete possível antes de migrarem para outra localidade.

Sabendo que, em um determinado período de tempo, N pessoas percorrem a trilha *Nunca-Mais*, calcule a quantidade máxima de energia que os canibais conseguirão obter ao capturar os andarilhos, considerando que este processo não pode exceder o custo máximo M aos canibais.

Entrada

A primeira linha é dada por pelos inteiros N ($1 \leq N \leq 10^4$), indicando a quantidade de pessoas que foram aventurar na trilha *Nunca-Mais* durante o referido período, e M ($1 \leq M \leq 10^4$), indicando o custo máximo que os canibais pretendem gastar no processo de captura de pessoas, separados por um espaço em branco.

A segunda linha contém N inteiros e_1, e_2, \dots, e_N , separados por um espaço em branco, descrevendo a energia que cada pessoa forneceria se for devorada pelos canibais, onde $1 \leq e_i \leq 10^4$.

A terceira linha também contém N inteiros c_1, c_2, \dots, c_N , separados por um espaço em branco, representando o custo de capturar cada pessoa, com $1 \leq c_i \leq 10^4$.

Saída

Imprima um único inteiro indicando a quantidade máxima de energia que os canibais podem obter após devorar as pessoas que eles conseguem capturar.

Exemplo

IV Maratona de Programação do IFB

Entrada	Saída
4 4	6
2 3 4 6	
3 1 8 4	
5 9	14
5 1 5 2 4	
2 6 4 3 3	

Notas

No primeiro caso de teste, os canibais obtêm a energia máxima ao capturar e devorar a pessoa 4, pois o custo para capturá-la não excede o custo que os canibais podem demandar nessa tarefa. É importante notar que os canibais também tem condições de capturar as pessoas 1 e 2, mas ao serem devoradas juntas pelos canibais, não oferecerão mais energia do que a pessoa 4.

No segundo caso de teste, os canibais conseguem obter a energia máxima ao devorar as pessoas 1, 3 e 5, pois juntas oferecem a quantidade de energia que respeita o custo máximo dos canibais no processo de captura.

Problema G

Canetta

Limite de tempo: 3s

Autor: Lucas Vasconcelos Mattioli

Marlin faz parte do melhor laboratório de software do Brasil, o CANETTA. No CANETTA, as pessoas são muito tímidas: sempre que chegam ao laboratório, só cumprimentam (com um aperto de mão) aqueles que consideram muito íntimos, deixando todo o resto do pessoal no vácuo. Na saída, a situação é ainda pior, ninguém se despede. Como as pessoas entram e saem de lá o tempo todo, situações constrangedoras são muito frequentes...

Formalmente, existem duas operações que uma pessoa i pode fazer: entrar ou sair do laboratório. Cada pessoa i possui uma lista de amigos L_i . Quando a pessoa x entra no laboratório, ela cumprimenta uma pessoa y somente se y estiver dentro do laboratório naquele momento e y estiver na lista de amigos L_x . A relação de amizades não necessariamente é bidirecional; isto é, pode ser que uma pessoa x esteja na lista de amizades de y , mas y não esteja na de x (é a vida). Cada pessoa é identificada unicamente por um inteiro entre 1 e N .

Seu trabalho é: dadas Q operações de entrada/saída de pessoas do laboratório e as listas de amizade de cada pessoa, contabilizar quantos apertos de mão (cumprimentos) cada pessoa realizou no total. É importante notar que no início do expediente não há funcionários no CANETTA e que quando x entra no laboratório e cumprimenta y , então o aperto de mão é contabilizado para x e para y .

Entrada

A primeira linha contém um inteiro N , $1 \leq N \leq 10^5$, indicando a quantidade de pessoas que trabalham no CANETTA.

A seguir são fornecidas N linhas, e cada uma inicia com um inteiro S_i , indicando a quantidade de amigos próximos desta i -ésima pessoa; seguido de S_i inteiros distintos, representando os identificadores dos amigos da i -ésima pessoa. É garantido que uma pessoa não faz parte da sua própria lista de amizade e que $0 \leq \sum_{i=1}^N S_i \leq 10^5$.

A próxima linha contém um inteiro Q , indicando a quantidade de operações de entrada/saída de pessoas no laboratório. Cada uma das próximas Q linhas, $1 \leq Q \leq 2 \cdot 10^5$, contém um inteiro entre 1 e N e um caractere em $\{E, S\}$, indicando, respectivamente, o identificador da pessoa realizando a operação e a operação em si: 'E' para entrada e 'S' para saída.

Saída

Ao final de todas as operações, imprima N inteiros em uma linha: o i -ésimo inteiro representa a quantidade de apertos de mão que a i -ésima pessoa realizou durante todo o conjunto de operações.

Exemplo

Entrada	Saída
3	1 1 0
2 2 3	
1 1	
2 1 2	
2	
1 E	
2 E	
4	2 1 0 3
1 4	
2 3 4	
1 1	
2 2 1	
6	
1 E	
4 E	
2 E	
1 S	
3 E	
1 E	
5	0 0 0 0 0
0	
1 1	
1 1	
1 1	
1 1	
5	
2 E	
3 E	
5 E	
4 E	
1 E	

Notas

No primeiro caso, o único aperto de mão acontece entre a pessoa 1 e a pessoa 2.

No segundo caso, os apertos acontecem entre os pares 1-4, 2-4 e 1-4.

Note que no último caso nenhum aperto de mão acontece, porque, apesar de todos considerarem a pessoa 1 como amiga, a pessoa 1 considera ninguém como amiga; assim, ao chegar no laboratório, ela cumprimenta ninguém.

Problema H

Inventário

Limite de tempo: 1s

Autor: Edson Alves da Costa Júnior

Paulo trabalhava no inventário de sua empresa quando encontrou uma nota de compra de mobília. Nesta nota constava o valor total da compra V , o preço unitário de uma mesa M e o preço unitário de uma cadeira C . Contudo, não estavam discriminadas quantas mesas x e quantas cadeiras y foram adquiridas.

Como o sistema de controle de inventário demanda ambas informações, escreva um programa que as determine, onde o valor de x deve ser o menor possível. Assuma que ao menos uma mesa e uma cadeira foram adquiridas.

Entrada

A entrada consiste em uma linha com os valores M , C e V ($1 \leq M, C \leq 10^9, 1 \leq V \leq 10^{18}$), separados por um espaço em branco.

Saída

Imprima, em uma linha, os valores x e y , separados por um espaço em branco. Se não for possível determinar um par de valores x e y que atenda as condições apresentadas, imprima o valor -1.

Exemplo

Entrada	Saída
500 100 3000	1 25
5000 200 3000	-1
123 45 6789	8 129

Notas

No primeiro caso, foram adquiridas 25 cadeiras e uma mesa. Veja que

$$1 \times 500 + 25 \times 100 = 3000$$

Embora a solução $x = 2$ e $y = 20$ leve ao mesmo total, não é aceitável pois x não tem o valor mínimo.

No segundo caso, não é possível comprar uma mesa, que custa 5000, e gastar apenas 3000. Logo, não há solução.

No terceiro caso, foram adquiridas 8 mesas e 129 cadeiras.

Problema I

AC ou WA?

Limite de tempo: 1s

Autor: Edson Alves da Costa Júnior

Um dos problemas do *contest* que Tarcísio participou era o de determinar o i -ésimo caractere da string de Fibonacci F_∞ , onde

$$\begin{aligned}F_1 &= \text{"B"} \\F_2 &= \text{"A"} \\F_n &= F_{(n-1)}F_{(n-2)}, \quad n > 2,\end{aligned}$$

onde a expressão $F_{(n-1)}F_{(n-2)}$ significa a concatenação das últimas duas strings de Fibonacci. Por exemplo, $F_3 = \text{"AB"}$, $F_4 = \text{"ABA"}$ e $F_5 = \text{"ABAAB"}$.

Conversando com os colegas após o encerramento do evento, Tarcísio percebeu que cometeu um equívoco na implementação: ele concatenou as strings em ordem oposta à definição, isto é, fez $F_{(n-2)}F_{(n-1)}$, $n > 2$. Esta modificação produzia strings S_i , não necessariamente iguais a F_i . Por exemplo, $S_3 = \text{"BA"}$, $S_4 = \text{"ABA"}$ e $S_5 = \text{"BAABA"}$.

Para determinar o i -ésimo caractere de F_∞ , ele gerou a string S_k de menor tamanho m tal que $m \geq i$ e retornou seu i -ésimo caractere, sendo que os índices são contados de 1 a m .

O que o deixou surpreso foi o fato do código errado ter passado pelos pré-testes do evento, ou seja, o código errado produziu as respostas corretas para estes testes. O veredito final do problema ainda não saiu, e Tarcísio está na dúvida: receberá um AC ou um WA?

Dado os índices que serão usados nos testes secretos, determine este veredito.

Entrada

A primeira linha da entrada contém o número N ($1 \leq N \leq 10^5$) de índices que serão usados nos testes secretos.

A segunda linha contém N inteiros positivos i ($1 \leq i \leq 10^{18}$), separados por um espaço em branco, representando os índices que fazem parte dos testes secretos.

Saída

Se o código de Tarcísio concordar com o gabarito em todos os índices presentes nos testes secretos imprima, em uma linha, a mensagem "AC". Caso contrário, imprima "WA".

Exemplo

IV Maratona de Programação do IFB

Entrada	Saída
3	AC
1 3 7	
5	WA
1 2 3 4 5	
4	AC
3 6 7 11	

Notas

No primeiro caso, temos $F_1 = \text{"B"} = S_1$ e $F_4 = \text{"ABA"} = S_4$. Além disso, $F_6 = \text{"ABAABABA"} = S_6$ e $S_6 = \text{"ABABAABA"} = S_6$, de modo que o sétimo caractere de ambas são iguais a 'B'. Assim, o veredito é **AC**.

No segundo caso, embora exista a igualdade no primeiro caractere, temos $F_3 = \text{"AB"} \neq \text{"BA"} = S_3$. O segundo caractere de ambas strings diferem ('B' \neq 'A'), logo o veredito é **WA**.

No terceiro, é possível ver que os códigos coincidem também no índice 6, ao observar as strings F_6 e S_6 apresentadas no primeiro caso. Para computar o décimo primeiro caractere, são utilizadas as strings $F_7 = \text{"ABAABABAABAAB"} = S_7$ e $S_7 = \text{"BAABAABABAABA"} = S_7$ e novamente, há a coincidência dos resultados. Logo o veredito é **AC**.

Problema J

Vendedor de Alpiste Neurótico

Limite de tempo: 1s

Autor: Jeremias Moreira Gomes

Depois do sucesso como vendedor de livros, Monk foi requisitado para ajudar na *International Fair of Birdseeds* (IFB) vendendo diferentes tipos de alpiste para pássaros.

Dessa vez Monk organiza criteriosamente os tipos de alpiste em potes de uma bancada. Por conta de seu Transtorno Obsessivo Compulsivo (TOC), esses potes são organizados em linha e cada coluna contém um pote com um tipo diferente de alpiste.

Para manter a bancada balanceada, Monk precisa que todos os potes tenham o mesmo peso. Sempre que vendas são realizadas, alguns potes ficam mais/menos pesados que outros. Por causa do TOC, sempre que uma rodada de vendas termina, Monk reorganiza a bancada repondo potes mais leves com alpiste do estoque ou retirando dos potes mais pesados de forma que, ao terminar de reorganizar, todos os potes tenham o mesmo peso.

Por querer tudo balanceado com o menor esforço possível, ele pediu a sua ajuda para saber, dados os pesos de cada um dos potes na bancada, qual o peso mínimo de alpiste deve ser retirado ou repostado para que todos os potes da bancada fiquem com o mesmo peso de alpiste.

Entrada

A entrada consiste de um único caso de teste. A primeira linha contém um valor inteiro $1 \leq N \leq 10^5$, que descreve a quantidade de potes. A segunda linha contém N números c_i (precisão dupla), que expressam os pesos de cada pote após uma rodada de vendas, com $0.1 \leq c_i \leq 10^8$ e $1 \leq i \leq N$.

Saída

A saída deve conter um único valor que representa o peso mínimo que deve ser retirado ou repostado, para que Monk consiga balancear os potes com o menor esforço possível.

Para cada caso de teste, se sua resposta é y e a resposta do juiz é z , sua resposta será considerada correta se $\frac{|y - z|}{\max(1, z)} \leq 10^{-3}$.

Exemplo

IV Maratona de Programação do IFB

Entrada	Saída
6	9.900000
1.001 2.101 3.201 4.301 5.401 6.501	
6	6.000022
1.210 3.210 3.210 5.210 3.210 1.210	
7	21.070007
2.010 2.020 9.030 2.040 9.050 9.060 4.070	

Notas

Para o primeiro caso, o balanceamento consiste em repor 2.200 no primeiro pote, repor 1.100 no segundo pote, repor 0.000 no terceiro pote, remover 1.100 do quarto pote, remover 2.200 do quinto pote e remover 3.300 do sexto pote.

Problema K

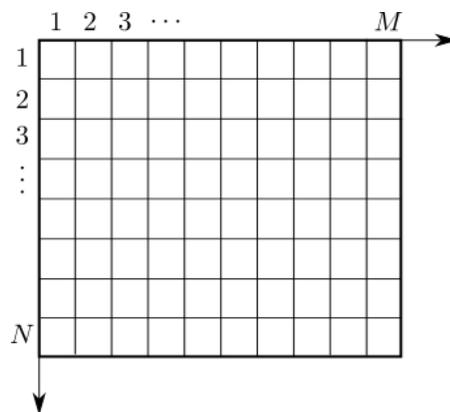
Campo Minado

Limite de tempo: 1s

Autor: Daniel Saad Nogueira Nunes

As províncias da Nlogônia e da Ackerland estão em guerra por conta de uma disputa envolvendo uma região repleta de minérios preciosos.

Esta região é um espaço retangular que contém N linhas e M colunas. As linhas são numeradas de 1 a N , considerando a orientação norte-sul e as colunas são numeradas de 1 a M , de acordo com a orientação oeste-leste, conforme figura abaixo:



A terra de Ackerland populou esta região com minas terrestres e, de tempos em tempos, os soldados colocam mais minas em posições específicas desta região de disputa.

Para não ficar para trás, o povo de Nlogônia está tentando desenvolver um poderoso detector de minas e você foi contratado como consultor pelo QG de Nlogônia para ajudá-lo.

Entrada

A primeira linha da entrada contém os valores dos inteiros N e M ($1 \leq N, M \leq 1000$), separados por um espaço em branco.

As próximas N linhas contém M valores inteiros V_{x_i, y_i} ($0 \leq V_{x_i, y_i} \leq 50$), separados por espaço em branco, cada qual correspondendo ao número de minas na posição (x_i, y_i) da região.

A próxima linha é descrita por um inteiro Q ($1 \leq Q \leq 10^5$), que indica o número de movimentos por parte dos povos de Nlogônia e Ackerland.

Cada uma das Q linhas seguintes está disposta cronologicamente, representa uma ação organizada por um dos lados e possui o seguinte formato:

- $0 X_1 Y_1 X_2 Y_2$: Nlogônia verifica o número de minas na região delimitada pelo retângulo com coordenada superior esquerda (X_1, Y_1) e coordenada inferior direita (X_2, Y_2) ($1 \leq X_1 \leq X_2 \leq M, 1 \leq Y_1 \leq Y_2 \leq N$).

- 1 X_1 Y_1 V : Ackerland coloca V ($1 \leq V \leq 50$) minas extras na posição (X_1, Y_1) da região.

Saída

Para cada uma das ações realizadas pela província de Nlogônia, o seu programa deverá imprimir o número de minas delimitado pelo retângulo com coordenadas (X_1, Y_1) e (X_2, Y_2) .

Exemplo

Entrada	Saída
3 3	45
1 2 3	16
4 5 6	32
7 8 9	49
5	
0 1 1 3 3	
1 2 2 4	
0 1 1 2 2	
0 2 2 3 3	
0 1 1 3 3	

Notas

No único caso de teste de exemplo, temos um campo minado com dimensões 3×3 e cinco consultas.

1. A primeira operação quer saber quantas minas existem no retângulo com coordenadas $(1, 1)$ e $(3, 3)$, que corresponde a soma de 1 a 9 e portanto equivale a 45.
2. A segunda operação adiciona 4 minas na posição $(2, 2)$ do campo.
3. A terceira operação quer saber quantas minas existem no retângulo com coordenadas $(1, 1)$ e $(3, 3)$, que corresponde a soma de $1 + 2 + 4 + 9 = 16$.
4. A quarta operação é uma consulta sobre o retângulo de coordenadas $(2, 2)$ e $(3, 3)$, cuja soma é $9 + 6 + 8 + 9 = 32$.
5. A quinta e última operação é uma consulta sobre o retângulo de coordenadas $(1, 1)$ e $(3, 3)$, cuja soma corresponde agora a $1 + 2 + 3 + 4 + 9 + 6 + 7 + 8 + 9 = 49$.

Problema L

Senha

Limite de tempo: 2s

Autor: Daniel Saad Nogueira Nunes

O jogo *Mastermind*, conhecido no Brasil como *Senha*, foi extremamente popular entre as décadas de 70 e 90 em todo o mundo.

Neste jogo existem 6 pinos de cada uma das 6 diferentes cores, um jogador desafiante e o jogador desafiado. O desafiante deve escolher uma senha secreta que consiste de 4 pinos e que pode conter pinos de cores repetidas. A cada rodada, o desafiado tenta adivinhar a senha com uma sequência de 4 pinos. O desafiante analisa a tentativa do desafiado e, se incorreta, fornece para ele uma dica: um par (p, q) , em que p corresponde ao número de pinos com cores corretas nas posições corretas e q o número de pinos com cores corretas em posições incorretas.

Esta dica dá prioridade para os pinos da tentativa de cor correta que estão na posição correta e considera cada pino da senha secreta uma única vez. Por exemplo: se a senha secreta é *(vermelho, azul, azul, preto)* e a tentativa é *(azul, azul, branco, azul)*, o desafiante retornará o par $(1, 1)$.

O desafiado ganha o jogo se acerta a senha em, no máximo, 10 tentativas. Caso contrário, o desafiante sai vencedor. O jogo é encerrado na primeira ocorrência de uma destas condições.

Seu objetivo é construir um programa que participe deste jogo onde o desafiante é o computador e que ganhe a maioria dos jogos.

Entrada

A entrada padrão deverá ser utilizada para ler as informações do desafiante.

Inicialmente o desafiante irá indicar um número ímpar N ($1 \leq N \leq 1000$) com a quantidade total de jogos a serem disputados.

A cada tentativa do usuário, o desafiante irá responder com uma linha contendo um par de inteiros p e q ($0 \leq p, q \leq 4$) separados por um espaço. Este par de inteiros compõem a dica descrita no enunciado.

O juiz só permitirá a leitura da dica depois da ocorrência de uma tentativa seguida de um *flush* da saída padrão.

Saída

Para realizar uma tentativa, a saída padrão deverá ser utilizada em conjunto com uma operação de *flush* para cada escrita.

O seu programa deverá imprimir uma tentativa, que consiste de uma linha contendo 4 inteiros $x_1x_2x_3x_4$ indicando os pinos escolhidos. Cada inteiro x_i pode assumir um valor entre 1 e 6, que indica a cor do i -ésimo pino.

Após a escrita da tentativa na saída padrão, o desafiante responderá com a dica na entrada padrão.

Interação

A implementação de uma solução “teimosa” em C++, onde o participante acredita que todas as senhas são iguais a "5555", seria

```
#include <bits/stdc++.h>

using namespace std;

int main() {
    int N;
    cin >> N;

    while (N--) {
        for (int i = 0; i < 10; ++i) {
            string guess = "5555";

            cout << guess << endl;
            fflush(stdout);

            int p, q;
            cin >> p >> q;

            if (p == 4 and q == 0)
                break;
        }
    }
    return 0;
}
```

Exemplo

Entrada	Saída
1	1234
0 3	3241
0 3	4312
1 2	3412
0 3	4123
2 0	4623
2 1	4125
4 0	

Notas

IV Maratona de Programação do IFB

No exemplo dado, temos que o total de um jogo será jogado. Temos a seguinte relação de tentativas/dicas do desafiado e desafiante:

desafiado	desafiante
1234	0 3
3241	0 3
4312	1 2
3412	0 3
4123	3 0
4623	2 0
4125	4 0

Ao todo, foram necessárias 8 tentativas para o desafiado acertar a senha, portanto ele ganha este jogo. Como a maioria dos jogos foram vencidas (havia apenas um), a resposta é aceita.

Problema M

Quadratura do Círculo

Limite de tempo: 1s

Autor: Edson Alves

A quadratura do círculo foi proposta pelos gregos e consistia em determinar, usando apenas uma régua sem marcações e um compasso, um quadrado cuja área fosse igual a área de um círculo dado.

Com a prova, no final do século 19, de que a constante π era transcendental, ficou provada a impossibilidade da solução do problema com as restrições apresentadas. Contudo, é possível aproximar as medidas de tal quadrado com o auxílio de computadores.

Dado o raio R , determine a medida do lado L do quadrado cuja área é igual a área do círculo delimitado por este raio.

Entrada

A entrada consiste em uma única linha, com o valor do inteiro R ($1 \leq R \leq 10^6$).

Saída

Imprima, em uma linha, a medida do lado L . Se a área do círculo é x e a área do quadrado de lado L é y , a resposta será aceita se $\frac{|x-y|}{\max(1,y)} < 10^{-6}$.

Exemplo

Entrada	Saída
2	3.544907701811
3	5.317361552717
5	8.862269254528

Problema N

Estraga-Festa

Limite de tempo: 1s

Autor: Daniel Saad Nogueira Nunes

A **IV Maratona de Programação do IFB** segue os moldes do ICPC, isto é: ganha quem tiver mais acertos e, em caso de empate, menor tempo acumulado. Cada submissão aceita da equipe é representada por um balão. Já o tempo acumulado funciona da seguinte maneira: se um time realiza uma submissão para um problema aos x minutos de competição e acerta o problema, adiciona-se $x + p \cdot 20$ ao tempo acumulado da equipe, em que p é o número de submissões incorretas anteriores para o problema considerado. Em caso de empate no número de balões e no tempo acumulado, ambas equipes dividem a mesma posição no placar.

A cerimônia de apresentação do placar final é sempre tensa, não se sabe o resultados das submissões enviadas nos últimos 60 minutos. As submissões são avaliadas uma a uma, e as alterações no placar apresentadas com estilo no *Boca Animeitor*. Não é raro que haja muitas mudanças no resultado final em funções de acertos na última hora, e os competidores acompanham o processo com muita atenção.

Filomena, Godofredo e Epaminondas são programadores de altíssimo nível e resolveram montar uma equipe chamada “Estraga-Festa” para participar da IV Maratona de Programação do IFB. O intuito deles é deixar a competição com um grau de emoção nunca visto antes. O plano do trio é enviar todas as soluções uma única vez e no minuto final da competição (minuto 300) para causar um ar de suspense gigantesco ao “rodar” o *Boca Animeitor*.

Sabendo da estratégia da equipe de antemão, você quer calcular se a equipe pode ou não estragar a festa dos demais competidores.

Entrada

A primeira linha da entrada possui dois inteiros, separados por espaço, N ($1 \leq N \leq 100$) e M ($0 \leq M \leq 20$), que representam respectivamente o número de equipes participantes, além da Estraga-Festa, e o número de submissões realizadas pela última.

Em seguida há N linhas, cada qual com um par de inteiros B_i ($0 \leq B_i \leq 20$) e T_i ($0 \leq T_i \leq 10^9$), separados por espaço, indicando respectivamente o número de balões adquiridos e o tempo acumulado da i -ésima equipe.

Considere que todas as equipes, excetuando a Estraga-Festa, não tem submissões pendentes.

Saída

Seu programa deverá imprimir ‘S’, caso exista a possibilidade da equipe Estraga-Festa ganhar a competição e ‘N’, caso contrário.

Exemplo

IV Maratona de Programação do IFB

Entrada	Saída
1 0	S
0 0	
3 3	N
5 905	
3 550	
4 440	
3 10	S
10 9001	
0 0	
8 800	

Notas

No primeiro caso de teste, a equipe Estraga-Festa não fez nenhuma submissão e a outra equipe não acertou nenhum problema, portanto a resposta esperada é 'S', pois houve empate entre as duas equipes.

No segundo caso de teste, mesmo que a equipe Estraga-festa acerte todos os problemas submetidos por ela, não ultrapassará a equipe que acertou 5 problemas, portanto a resposta esperada é 'N'

No terceiro caso de teste, caso a equipe Estraga-festa acerte todas as submissões, ela terá 10 balões e um tempo acumulado menor do que a equipe melhor colocada, que também possui 10 balões. Assim, a resposta esperada é 'S'

Problema O

Construindo Genomas

Limite de tempo: 3s

Autor: Daniel Saad Nogueira Nunes

Tyrannosaurus é um gênero de dinossauro cujo representante mais conhecido é o *Tyrannosaurus rex*. Uma questão em aberto na Paleontologia é determinar se o *rex* é ou não a única espécie do gênero *Tyrannosaurus*. Alguns acreditam na tese de que o *Tyrannosaurus bataar* seja uma nova espécie, já outros acreditam que os fósseis desta suposta nova espécie na verdade se tratam de fósseis de *T. rex*. Outras espécies também foram propostas, mas foram desconsideradas pois se enquadravam em uma das duas possíveis espécies já mencionadas.

Dr. Ross Unagi, um famoso paleontólogo da cidade de Nova Iorque, acredita que um fóssil recém descoberto não se enquadra como *T. rex* ou *T. bataars*, mas sim como *Tyrannosaurus sufixi*. Para verificar se o fóssil encontrado representa uma nova espécie, o Dr. Unagi quer reconstruir o genoma a partir do material biológico do fóssil. No entanto há um problema: os sequenciadores de DNA não conseguem obter o genoma inteiro a partir do material biológico, apenas fragmentos.

O Dr. Unagi pediu a sua ajuda para obter o genoma a partir do fóssil. Este genoma tem as seguintes propriedades:

- os fragmentos são suficientes para obtenção do genoma;
- cada fragmento aparece pelo uma vez no genoma; e
- o tamanho do genoma é o menor possível.

Baseando-se nessas propriedades, será que você consegue juntar os fragmentos e reconstruir o genoma? Ajude o Dr. Unagi a montar este quebra-cabeça!

Entrada

A entrada consiste de uma linha com um inteiro N ($1 \leq N \leq 11$) representando o número de fragmentos produzidos pelo sequenciador de DNA.

As próximas N linhas possuem, cada uma, um fragmento de DNA, que consiste de uma sequência sobre os símbolos $\{a, c, t, g\}$. Cada fragmento é composto por, no máximo, 50 símbolos.

Saída

Seu programa deverá imprimir uma linha com o tamanho do genoma encontrado e o genoma propriamente dito. Na possibilidade de múltiplos genomas atenderem as propriedades descritas, qualquer um pode ser utilizado como resposta.

Exemplo

IV Maratona de Programação do IFB

Entrada	Saída
3 actg gctac tact	7 gctactg
3 acg tacgt ttat	8 tacgttat
3 atgc gtct ctc	9 atgcgtctc

Notas

No primeiro caso de teste uma possível resposta é o genoma “gctactg”, uma vez que todos os fragmentos podem ser encontrados nele e ele tem tamanho mínimo.

O mesmo se aplica ao segundo e terceiro casos de testes. Uma observação que pode ser feita é que, no terceiro caso de teste, o genoma “gtctcatgc” também configura uma resposta válida.