Seletiva UnB 2025

Caderno de Problemas

19 de julho de 2025

(Este caderno contém 14 problemas)

Comissão Organizadora:
Alberto Tavares Duarte Neto (UnB)
Arthur Botelho (UnB)
Daniel Porto (UnB)
Daniel Saad Nogueira Nunes (IFB)
Edson Alves da Costa Júnior (UnB/FCTE)
Eduardo Freire (UnB)
Guilherme Novaes Ramos (UnB)
Guilherme Rocha (UnB)
Jeremias Moreira Gomes (IDP)
José Marcos da Silva Leite
Pedro Gallo (UnB)
Ruan Petrus (UnB)

Patrocínio:

Universidade de Brasília, *campus* Gama

Lembretes

- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova, entretanto, o mesmo não vale para materiais dispostos eletronicamente.
- 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;
- Para cada problema, além dos testes públicos, o juiz executará a sua submissão contra uma série de testes secretos para fornecer um parecer sobre a correção do programa.
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. Lembre-se que as soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Utilize a aba *clarification* para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos;

C/C++

- Seu programa deve retornar zero, executando, como último comando, `return 0` ou `exit 0`.

Java

- Não declare `‘package’` no seu programa Java.
- Note que a conveção para o nome do arquivo fonte deve ser obedecida, o que significa que o nome de sua classe pública deve ser uma letra maiúscula igual a letra que identifica o problema.

Python

- Tenha cuidado ao selecionar a versão correta na submissão.

Dedicatória

Grupo MaratonasDF

19 de julho de 2025

Com profunda gratidão e reconhecimento, dedicamos este caderno de problemas e a seletiva UnB 2025 ao Professor Vinicius Ruela Pereira Borges por suas contribuições ao cenário da programação competitiva no Distrito Federal.

Vinicius tomou as rédeas das atividades no Campus Darcy Ribeiro em 2018 e ajudou a elevar a UnB de um grupo de entusiastas talentosos para uma incrível abordagem fordista de capacitação de competidores de alto desempenho. Incansável em suas atividades, sua presença como técnico é garantia de Mundial e suas fotos imortalizaram as atividades que hoje destacam o DF no cenário de programação competitiva nacional. Seu impacto nas disciplinas de apoio, grupos de estudo e atividades de extensão e igualdade de gênero é incomparável. Destaque para a revolução na indústria de coffee-breaks com café de qualidade e na excelente fotografia dos eventos, dando ao pão de queijo a devida resolução de 4K.

Sua dedicação foi tamanha que nunca se entendeu como ele conseguia aparecer em todos os eventos, responder mensagens no grupo às 3h da manhã, passar a madrugada em claro nos eventos de 24 horas, e ainda parecer mais descansado que os próprios alunos. Há boatos de que ele foi clonado em laboratório por uma IA altamente motivada com café e gratificação simbólica. Se é verdade? Não sabemos. Mas se alguém um dia descobrir como fazer uma versão em Docker do Vinicius, por favor, compartilhe.

E agora, enquanto se despede temporariamente da maratona para fazer o pós-doutorado (a versão "*new game plus*" da academia), nos resta torcer para que os pesquisadores internacionais estejam preparados para lidar com alguém que resolve um problema difícil antes mesmo de terminar de ler o enunciado. Que eles tenham bom Wi-Fi, café decente e nervos tão firmes quanto os nossos durante a última hora de uma regional.

Agradecemos por tudo,

Grupo MaratonasDF.

PS: *"Vuelve Vini!"* [M. A. R., 2025]

Problema A – Athleta Alemão

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Guilherme Ramos

Vin Borgen é um alemão radicado em Brasília, que abandonou seu cotidiano de *weißwurst und weißbier* para forjar um glorioso histórico de athleta no interior de Minas Gerais. Multicampeão de corridas, ele treina intensamente para manter o ritmo (e as vitórias). É fácil reconhecê-lo nestes momentos: tênis de corrida colorido, shorts curto, óculos escuros e um boné. Claro, sempre com sua camiseta de coach da Maratona SBC de Programação[1].

Herr Borgen sabe que o segredo do sucesso alemão é a *laufschrittfrequenz*, a cadência da volta do athleta. Cada volta no circuito deve ser feita dentro de um intervalo de tempo razoável para a sua cadência, de modo a manter um desempenho ótimo e constante. Ajude-o a acompanhar e ajustar seu treino!

Entrada

A primeira linha fornece a meta do athleta alemão — o tempo de uma volta na cadência adequada para o circuito, em minutos. A seguir é dada uma quantidade indeterminada de linhas, cada uma indicando o tempo, em minutos, de uma volta dada por *Herr* Borgen (nenhuma volta demora menos de 1 minuto ou mais de 12 horas). Ele corre até não conseguir manter o ritmo.

Saída

Para cada volta, apresente uma mensagem de apoio ao treino do athleta. Se desviar mais que 5% da meta, diga "Athelera, fera!" (se for lento) ou "Calma que ainda tem o returno." (se for rápido). Caso contrário, apresente "Boa! Merece cafe e pao de queijo."

Exemplo

Entrada	Saída
`50` `50`	`Boa! Merece cafe e pao de queijo.`

`60`
`62`
`63`
`64`
	`Boa! Merece cafe e pao de queijo.`
	`Boa! Merece cafe e pao de queijo.`
	`Athelera, fera!`

Notas

[1] *Herr* Borgen tem 100% de aproveitamento em classificação para mundiais! Ele é conhecido no submundo como *hopfentrainer* (treinador saltador).

Problema B – Balões de Festa

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Arthur Botelho

O professor Vinicius, bastante querido na UnB, dá aula em N turmas, numeradas de 1 a N. Para a sua despedida, as N turmas decidiram organizar N festas surpresas para o professor Vinicius, com bolo, salgadinhos e balões de festa.

Cada turma i comprou B_i balões para a sua festa surpresa. No entanto, os alunos decidiram que seria melhor se todas as festas tivessem a mesma quantidade de balões, para ser justo. Assim, cada turma deve descartar alguma quantidade de balões (possivelmente 0), de forma que todas fiquem com a mesma quantidade de balões ao final. Por exemplo, se há 2 turmas com 3 e 4 balões, respectivamente, a primeira turma pode descartar 2 balões e a segunda 3, de forma que ambas ficam com 1 balão ao final.

Os alunos querem evitar desperdício — os balões são caros! Por isso, diga a eles qual é a menor quantidade total de balões que precisam ser descartados pelas turmas para que todas fiquem com a mesma quantidade de balões ao final.

Entrada

A primeira linha da entrada contém um único número N ($1 \leq N \leq 100$), a quantidade de turmas.

A segunda linha da entrada contém N inteiros B_i ($0 \leq B_i \leq 100$), em que B_i é a quantidade de balões comprados pela turma i.

Saída

Imprima uma única linha contendo um único inteiro: a menor quantidade total de balões que precisam ser descartados pelas turmas para que todas fiquem com a mesma quantidade de balões ao final.

Exemplo

Entrada	Saída
`2` `3 4`	`1`
`4` `0 0 0 0`	`0`
`3` `8 5 100`	`98`

Notas

No primeiro caso de exemplo, a melhor opção é a primeira turma descartar 0 balões, a segunda descartar 1, ficando ambas com 3 balões.

No segundo caso de exemplo, como nenhuma turma trouxe balões, não será necessário descartar nada.

No terceiro caso de exemplo, a melhor opção é a primeira turma descartar 3, a segunda descartar 0 e a terceira descartar 95.

Problema C – Cafezim

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Daniel Porto

O professor Vinícius, bom mineiro que é, adora um cafezim. Mas tem um problema sério: o café do departamento é tão ruim, mas tão ruim, que da última vez ele serviu em copo descartável, ele teve que mastigar o café. Parecia mais uma sopa de carvão feita com raiz do pé de café. Estava tudo torrado junto.

Indignado, Vinícius tomou uma decisão radical: passou a levar o próprio café, preparado com carinho, moagem na mão, e água mineral do alto da serra da canastra. Ele leva o café num barril. Não chega a ser um barril de chopp, mas é de respeito.

E como todo mineiro hospitaleiro, ele leva junto uma capanga cheia de pães de queijo quentinhos. O hábito virou tradição. Hoje tem até professor que anda com um pão de queijo mordido no bolso. Isso só pra garantir o cafezim, caso encontre com o Vinícius no corredor. A universidade nunca mais foi a mesma. Existe um antes e um depois do professor Vinícus. Mas aí surge um desafio logístico!

Como são muitos professores, cada um com uma xícara de tamanho diferente (tem desde caneca da Hello Kitty até copinho de cachaça da formatura de 78), Vinícius quer escolher exatamente k xícaras para encher com a mesma quantidade de café, e ele não quer ver uma gota derramada e nenhuma xícara transbordando.

Antes que a próxima fornada de pão de queijo esfrie, você deve ajudá-lo a descobrir qual é o maior volume inteiro de café x que pode ser colocado em k xícaras diferentes, de forma que:

- Cada xícara escolhida consiga suportar ao menos x mL;
- O volume total de café usado não ultrapasse o que ele trouxe no barril.

Entrada

A primeira linha contém três inteiros n, k e v indicando respectivamente: o número de professores (e xícaras), o número de xícaras que que o professor Vinícius deseja servir, e o volume total de café disponível no barril ($1 \leq k \leq n \leq 10^5, 1 \leq v \leq 10^9$).

A segunda linha contém n inteiros $c_1, c_2, ..., c_n$, onde c_i representa a capacidade da xícara i em mililitros ($1 \leq c_i \leq 10^9$).

Saída

Imprima o maior valor inteiro x tal que seja possível encher exatamente k xícaras, cada uma com exatamente x mL de café, sem transbordar nenhuma, e sem exceder o volume total disponível no barril. Caso não seja possível encher k xícaras com nenhuma quantidade de café, responda 0.

Exemplo

Entrada	Saída
5 3 400 50 70 52 10 100	52
4 2 100 100 200 300 400	50
10 5 1000 50 100 75 30 200 150 30 200 90 13	90

Problema D – Dias de Viagem

Limite de tempo: 4s
Limite de memória: 128MB

Autor: Eduardo Freire

Antes de ir para a Alemanha, o professor Vinícius decidiu visitar um dos países mais famosos entre os maratonistas: a Nlogônia. O professor selecionou um período de N dias consecutivos em que ele potencialmente realizaria sua viagem. Existem três cidades desse país que ele deseja visitar: Raizgônia (R), Quadragônia (Q) e Inversoackermangônia (I). No i-ésimo dia, existem exatamente RQ_i passeios turísticos distintos entre Raizgônia e Quadragônia, RI_i passeios entre Raizgônia e Inversoackermangônia e QI_i passeios entre Quadragônia e Inversoackermangônia. Todo passeio turístico pode ser realizado em ambos os sentidos, ou seja, se em um dado dia existem X passeios partindo da cidade A para a cidade B, também existem exatamente X passeios partindo de B e chegando em A.

Em cada dia de sua viagem, Vinícius quer participar **de no máximo um passeio**. Como as empresas que organizam os passeios turísticos são imprevisíveis e alteram a quantidade de passeios disponíveis em cada dia frequentemente, ele quer que você ajude-o a realizar Q ações, cada ação podendo ser de dois tipos:

- Tipo 1 — Serão dados caracteres $A, B \in \{R, Q, I\}$ e inteiros L e R ($1 \leq L \leq R \leq N$). Supondo que Vinícius estará na cidade A no dia L, calcule de quantas formas ele poderá realizar passeios ao longo dos dias de forma que ele termine na cidade B no dia R. Duas formas são consideradas distintas se em um dos dias Vinícius pegou um passeio diferente, ou se ele deixou de passear neste dia em uma forma e não na outra. Como a resposta pode ser grande, imprima-a módulo 998244353, ou seja, imprima o resto de sua divisão por 998244353.
- Tipo 2 — Dados caracteres $A, B \in \{R, Q, I\}$, um inteiro d ($1 \leq d \leq N$) e um inteiro x ($0 \leq x < 998244353$), altere a quantidade de passeios existentes entre as cidades A e B no d-ésimo dia para x. Nessa ação, é garantido que $A \neq B$.

Entrada

A primeira linha contém 2 inteiros N e Q ($1 \leq N, Q \leq 10^5$), representando a quantidade de dias e a quantidade de ações a serem realizadas.

Seguem N linhas, a i-ésima das quais contém três inteiros RQ_i, RI_i e QI_i ($0 \leq RQ_i, RI_i, QI_i < 998244353$), representando a quantidade de passeios turísticos existentes entre cada par de cidades distintas no i-ésimo dia.

Em seguida, existem Q linhas, a i-ésima das quais pode ter um dos dois formatos:

- 1 A B L R
- 2 A B d x

O significado e tamanho de cada um desses valores é descrito no enunciado.

Saída

Para cada ação do tipo 1, imprima um único inteiro por linha: o resto da divisão por 998244353 da quantidade de maneiras diferentes de realizar os passeios com a especificação dada.

Exemplo

Entrada	Saída
2 4 1 2 3 1 0 2 1 Q Q 2 2 1 I R 1 2 2 R I 2 4 1 R I 2 2	1 5 4
5 5 72 96 16 79 31 55 52 46 52 79 67 2 58 36 13 1 Q R 3 4 2 R I 3 81 1 I R 2 5 2 R Q 1 80 1 Q R 1 4	3615 22324879 43134751

Notas

No primeiro caso de teste, existe exatamente uma maneira de ir de Quadragônia para Quadragônia utilizando os passeios disponíveis no primeiro dia: basta que Vinícius não realize um passeio nesse dia. Para a segunda ação, Vinícius pode escolher um dos 3 passeios de Inversoackermangônia para Quadragônia no primeiro dia e no segundo dia realizar o único passeio de Quadragônia para Raizgônia. Também seria possível no primeiro dia ir diretamente para Raizgônia utilizando um dos 2 passeios disponíveis e não passear no segundo dia. Assim, a resposta para essa ação é $3 + 2 = 5$.

Problema E – Espera

Limite de tempo: 2s
Limite de memória: 256MB

Autor: Jeremias Moreira Gomes

De mala e cuia na mão, ou melhor dizendo, de mala e tênis na mão, Vinícius está na sala de espera do aeroporto esperando o horário do seu voo. O único problema é que ele chegou muito cedo e agora vai ter que ficar esperando por bastante tempo até o seu voo decolar.

Para passar o tempo, ele estava lendo as informações do seu cartão de embarque e percebeu que há um identificador único escrito nele com alguns caracteres repetidos. Então ele ficou curioso em saber, para cada posição, a quantidade de vezes que o caractere dessa posição aparece em outra posição. Depois de passar bastante tempo pensando, ele decidiu pedir a sua ajuda para que você escreva um programa que faça essa contagem.

Entrada

A entrada contém múltiplos casos de teste. A primeira linha contém um inteiro T ($1 \leq T \leq 1000$) que representa o número de casos de teste. Cada caso de teste contém uma linha com uma string S ($1 \leq |S| \leq 10000$), composta por números, letras maiúsculas ou letras minúsculas, que representa o identificador único do cartão de embarque.

Saída

Para cada caso de teste, imprima uma linha contendo um inteiro que representa a soma, considerando todas as posições i do identificador, da quantidade de vezes que o caractere na posição i aparece em outras posições.

Exemplo

Entrada	Saída
`1` `S97M4SLB`	`2`

`3`
`S313T1V4`
`ASCHAFFENBURG`
`OBERAMMERGAU`
	`4`
	`4`
	`8`

Problema F – Fura-Cão

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Guilherme Ramos

O Fura-Cão faz parte da família de mascotes do Club Athletico Paranaense (CAP), o famoso Rubro-Negro da Baixada. Autodenominado da raça "vira-taças", o cachorro tenta incentivar o time e incomodar os rivais.

Pereirão é grande entusiasta do time e do mascote, ele sempre carrega a pelúcia do Fura-cão quando vai acompanhar um jogo do Athletico. Ele vibra e xinga com todas as forças! Ajude-o nesta empreitada.

Entrada

A entrada consiste de uma linha descrevendo um acontecimento do jogo, sempre indicando qual o time envolvido.

Saída

Quando for CAP está em ação, Pereirão e Fura-cão celebram o time, mas quando o time rival está envolvido, eles não poupam xingamentos, conforme os exemplos.

Exemplo

Entrada	Saída
`La vem o athletico ameacando o gol!`	`E da-lhe Athletico!`
`Olha os coxa-brancas pressionando.`	`Grrrrr! CAP! CAP! CAP!`
`Furacao avanca com tudo!`	`E da-lhe Athletico!`
`E GOOOOOL DO ATHLETICO!`	`E da-lhe Athletico!`
`Gremio atacando pelo meio.`	`Grrrrr! CAP! CAP! CAP!`

Notas

O Club Athletico Paranaense é sempre facilmente identificado pela escrita "diferenciada" do seu nome ou pelo uso do seu principal apelido: Furacão.

Problema G – Guardadim

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Edson Alves

Vinícius leva a sério a importante questão de guardar e conservar seus queijos. Todo mundo sabe (ou deveria saber, uai!) que, para armazenar um queijo, é preciso observar a temperatura e a umidade do ambiente, para evitar ressecamento e contaminação. Esses parâmetros, naturalmente, dependem do tipo de queijo a ser armazenado (se é fresco ou duro, por exemplo).

Usando seu vasto conhecimento de computação e sistemas embarcados, Vinícius preparou um cômodo para armazenar seus queijos e colocou sensores na bancada principal. Estes sensores capturam, a cada instante, as informações do ambiente e da bancada e retornam ao servidor central, para cada centímetro quadrado, um coeficiente c entre 0 e 1.

Vinícius deseja armazenar seu novo lote de queijos em uma subárea retangular de sua bancada, a qual tem dimensões $H \times W$ cm^2 e é subdividida em quadrados de área 1 cm^2. Para o devido armazenamento dos queijos deste lote, a subárea selecionada deve ser tal que cada um dos quadrados que a compõem tenha coeficiente c entre A e B, inclusive.

Como o lote de queijos é grande, Vinícius precisa determinar a maior subárea retangular com a característica citada para deixar o queijo "bem guardadim...". Ele poderia fazer isso sozinho em uma fração de segundos, mas pediu para você resolver o problema enquanto ele confere se o lote é "bão mesmo" com um cafezim...

Entrada

A primeira linha da entrada contém as dimensões H e W ($2 \leq H, W \leq 500$) da bancada, separadas por um espaço em branco.

A segunda linha contém os números reais A e B ($0.000 \leq A \leq B \leq 1.000$), separados por um espaço em branco. Esses números serão informados com exatamente 3 casas decimais.

As próximas H linhas contêm W coeficientes c_{ij} ($0.000 \leq c_{ij} \leq 1.000$), separados por um espaço em branco, indicando o coeficiente do j-ésimo ($1 \leq j \leq W$) quadrado da i-ésima ($1 \leq i \leq H$) linha da malha de quadrados de área 1 cm^2 que compõe a bancada. Esses números também serão informados com exatamente 3 casas decimais

Saída

Se não existe nenhuma subárea retangular na bancada que satisfaça o critério apresentado, imprima a mensagem "`Nao`".

Caso contrário, imprima a mensagem "`Sim`". Na linha seguinte, imprima a maior área possível da bancada que pode ser usada para armazenar os queijos. Na terceira linha imprima as coordenadas a e b do canto superior esquerdo da maior área onde os queijos podem ser armazenados, separadas por um espaço em branco. Na quarta e última linha imprima as coordenadas c e d do canto inferior direito, separadas por um espaço em branco. Se houver mais de um retângulo que maximize a área total, imprima qualquer um deles.

Exemplo

Entrada	Saída

`3 3`
`0.500 0.750`
`0.489 0.673 0.800`
`0.510 0.711 0.545`
`0.789 0.621 0.750`
	`Sim`
	`4`
	`2 2`
	`3 3`

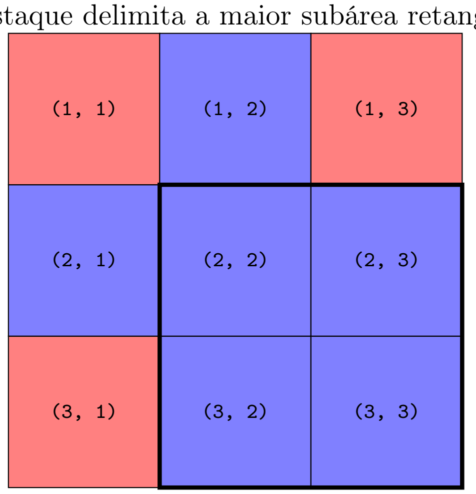
`2 2`
`0.999 1.000`
`0.989 0.998`
`0.979 0.991`
	`Nao`

`4 4`
`0.800 0.800`
`0.987 0.513 0.212 0.776`
`0.400 0.123 0.969 0.323`
`0.000 0.999 0.801 0.802`
`0.899 0.800 0.799 0.801`
	`Sim`
	`1`
	`4 2`
	`4 2`

`2 3`
`0.000 0.999`
`0.121 0.826 0.555`
`0.975 0.727 0.001`
	`Sim`
	`6`
	`1 1`
	`2 3`

Notas

A figura abaixo ilustra o primeiro caso: os quadrados azuis atendem ao critério e a linha em destaque delimita a maior subárea retangular cujos elementos todos atendem ao critério.



No segundo caso, nenhum quadrado atende ao critério estabelecido.
No terceiro caso, um único quadrado atende ao critério.
No quarto caso, todos os quadrados atendem ao critério.

Problema H – H-árvore

Limite de tempo: 1.5s
Limite de memória: 256MB

Autor: Alberto Neto

Vinicius e Guilherme estão jogando um jogo maluco, chamado H-árvore, e o perdedor terá que pagar a conta do almoço na próxima Nacional. O jogo é feito em uma árvore de n vértices numerados de 1 a N, e cada vértice i tem um valor a_i que pode ser 0 ou 1. No inicio do jogo, cada jogador recebe uma variedade de cartas, cada uma permitindo diferentes ações, e em cada turno um dos jogadores usa ou descarta uma de suas cartas. Depois de todos os jogadores usarem ou descartarem todas as suas cartas, o seguinte acontece:

- Seja e_{ij} uma aresta entre os vértices i e j. Se $a_i \neq a_j$, então a aresta e_{ij} é removida da árvore.
- Após todas as arestas serem consideradas, um complexo cálculo é feito para computar a pontuação de cada jogador.

Após uma intensa partida, os professores chegam no último turno, que é de Vinicius. Sua última carta o permite escolher qualquer vértice i e inverter seu valor: se $a_i = 0$ então seu valor passa a ser 1, mas se $a_i = 1$, então seu valor passa a ser 0. Note, porém, que Vinicius também pode **descartar** sua carta final. Nesta situação do jogo, Vinicius percebe que, para maximizar sua pontuação, deve **maximizar** o número de componentes conexos do grafo após o fim de jogo.

Ajude Vinicius a ganhar o jogo (e o almoço) e calcule a maior quantidade de componentes conexos possível ao fim do jogo, se utilizar sua carta final de maneira ótima.

Entrada

A primeira linha de entrada contém um único inteiro N ($2 \leq N \leq 2 \cdot 10^5$) — o número de vértices da árvore.

A segunda linha contém N inteiros $a_1, \ldots, a_N$ ($0 \leq a_i \leq 1$) — o valor a_i do vértice i no último turno do jogo.

Dentre as próximas $n-1$ linhas, a j-ésima delas contém dois inteiros x_j e y_j ($1 \leq x_j, y_j \leq N$; $x_j \neq y_j$) — representando que a j-ésima aresta conecta os vértices x_j e y_j. **É garantido que o grafo formado é uma árvore**.

Saída

Imprima uma linha contendo um único inteiro — a quantidade máxima de componentes conexos do grafo obtido ao fim do jogo após mudar o valor de até um vértice.

Exemplo

Entrada	Saída
3 0 0 0 1 2 2 3	3
3 0 1 0 1 2 2 3	3
7 1 0 1 0 1 0 0 1 2 3 1 4 2 1 5 2 6 1 7	4

Notas

Uma **árvore** T é um grafo não direcionado conexo de n vértices e $n-1$ arestas. Um **componente conexo** C de um grafo G é um subconjunto de vértices tal que, para todo par de vértices em C, é possível percorrer de um até outro usando as arestas de G, mas não é possível ir de um vértice de C até um vértice fora de C.

No primeiro caso de teste, se trocamos o valor do vértice 1, teremos dois componentes ao final do jogo: $\{1\}$ e $\{2,3\}$. A inversão do valor do vértice 3 também dará dois componentes. Porém, ao inverter o valor do vértice 2, teremos os componentes $\{1\}$, $\{2\}$ e $\{3\}$, o que maximiza a resposta.

No segundo caso de teste, o ótimo é não inverter nenhum valor.

Problema I – Ironman

Limite de tempo: 4s
Limite de memória: 256MB

Autor: Daniel Porto

Depois de anos enfrentando bancas de TCC, formulários da CAPES e o famigerado SEI, o Professor Vinícius finalmente decidiu encarar um desafio ainda mais extremo: participar da edição mundial do Ironman na Alemanha. Afinal, se é pra sofrer, que seja no frio da Europa e não só à beira do lago Paranoá.

Entre uma aula de programação competitiva e outra de natação no Lago Wannsee, nosso Ironman do cerrado decidiu começar a monitorar os seus batimentos cardíacos no seu smartwatch, afinal a temperatura estava "de cair os bits".

Os dados registrados formam uma sequência de inteiros $F_1, F_2, ..., F_N$, representando a frequência dos batimentos cardíacos do professor durante os treinamentos. Vinícius percebeu um padrão curioso nos dados do seu smartwatch. O gráfico da frequência cardíaca (em batimentos por minuto) formava uma sequência com picos e vales. Diante disto, ele quis saber qual foi a maior diferença positiva entre dois batimentos, desde que a maior tenha ocorrido depois da menor.

Vinícius suspeita que o pico aconteceu quando ele percebeu que tinha pedalado 20 km a mais porque confundiu uma placa em alemão com uma dica de percurso. Mas não tem certeza. Sua missão é ajudá-lo a calcular essa diferença máxima.

Entrada

A primeira linha contém um inteiro N ($2 \leq N \leq 10^6$) — o número de leituras feitas durante o treino.

A segunda linha contém N inteiros $F_1, F_2, \ldots, F_N$, tais que $0 \leq F_i \leq 10^5$ — a frequência cardíaca registrada em cada momento.

Saída

Imprima um único inteiro: a maior diferença $F_j - F_i$ tal que $j > i$. Se não for possível obter uma diferença positiva, imprima 0.

Exemplo

Entrada	Saída
8 80 82 79 75 85 70 90 60	20
4 100 200 300 400	300
5 100 75 30 30 13	0

Notas

Resolva o problema antes que a pouca bateria do smartwatch acabe. Quem sabe podemos usar o resultado pra gerar um gráfico no próximo artigo da revista "Computação & Suor".

Problema J – João Grilo

Limite de tempo: 1s
Limite de memória: 256MB

Autor: Edson Alves

Preocupado com o futuro e com o preço do café, o professor Borges investiu todas as suas economias (que totalizavam um real) numa carteira de *byte coins* (que sabidamente rendem 8 vezes mais do que as tradicionais *bit coins*...): a cada mês, o valor que ele tem investido é multiplicado por um fator K!

Contudo, de bobo ele não tem nada, e sabe que esta situação pode mudar a qualquer momento. Assim, ao final de cada mês, após a atualização de seu investimento, ele compra tantas barras de ouro quanto possível (também é fato notório que barras de ouro valem muito mais do que dinheiro...). Cada barra de ouro custa B reais e não é possível comprar frações de uma barra.

Por exemplo, se $K = 7$ e $B = 120$, teremos a seguinte situação:

1. Borges investe seu real;
2. No início do primeiro mês, ele terá 7 reais investidos;
3. No início do segundo mês, ele terá $7 \times 7 = 49$ reais investidos;
4. No início do terceiro mês, ele terá $7 \times 49 = 343$ reais investidos. Ele compra então duas barras de ouro por 240 reais e deixa os 103 reais restantes no investimento.

Uma coisa curiosa acontece, porém, no quarto mês: ele terá $103 \times 7 = 721$ reais investidos, de modo que ele comprará 6 barras de ouro por 720 e ficará novamente com apenas 1 real investido. Este fato o lembrou da frase clássica de seu herói João Grilo: "estou cansado desta agonia de ficar rico, ficar pobre, ficar rico, ficar pobre...".

Tranquilize (ou não) o professor Borges escrevendo um programa que recebe os valores de B e de K e que determine o primeiro mês em que ele, após comprar todas as barras de ouro possíveis, tornará a ter exatamente um real investido.

Entrada

A entrada é composta por uma única linha, contendo os valores dos inteiros K ($2 \leq K \leq 10^3$) e B ($3 \leq B \leq 10^9$), separados por um espaço em branco.

Saída

Imprima, em uma linha, o número do primeiro mês no qual o professor Borges, após comprar todas as barras de ouro possíveis, torna a ter exatamente um real investido.

Caso o professor jamais volte a ficar pobre (isto é, ele sempre terá mais do que um real investido), imprima o valor -1.

Exemplo

Entrada	Saída
`7 120`	`4`
`12 50`	`-1`
`11 12345678`	`335616`

Problema K – Kaos

Limite de tempo: 4s
Limite de memória: 512MB

Autor: Alberto Neto

Vinicius, em sua pesquisa na Alemanha, está estudando permutações **x-kaóticas**. Uma permutação $p_1, \ldots, p_n$ de tamanho n é dita x-kaótica, para um inteiro positivo x, se existem exatamente x posições da permutação p tais que $p_i \geq i$. Como Vinicius achou muito interessante o problema de calcular quantas permutações x-caóticas existem, ele submeteu o problema para a Seletiva UnB 2025.

Sua equipe, como participante da Seletiva, deve fazer um programa que calcule, para cada inteiro x de 1 até n, a quantidade de permutações x-kaóticas. Como esses valores podem ser muito grandes, calcule-os módulo 998244353.

Entrada

A única de entrada contém um inteiro n ($1 \leq n \leq 400$) — o tamanho n das permutações a serem consideradas.

Saída

Imprima n inteiros $k_1, k_2, \ldots, k_n$ — a quantidade de permutações x-kaóticas, para os inteiros x de 1 até n, em ordem. Esses valores devem ser calculados módulo 998244353.

Exemplo

Entrada	Saída
3	1 4 1
4	1 11 11 1
9	1 502 14608 88234 156190 88234 14608 502 1

Notas

Uma permutação é uma sequência de tamanho n contendo cada inteiro de 1 até n exatamente uma vez.

No primeiro caso de teste, a permutação $[1, 2, 3]$ é 3-caótica e a permutação $[3, 1, 2]$ é 1-caótica. Todas as demais permutações de 3 elementos são 2-caóticas.

Problema L – Legendary Coach

Limite de tempo: 4s
Limite de memória: 256MB

Autor: Alberto Neto

Com o sucesso do jogo UnBalloomon TCG, os desenvolvedores adicionaram um novo recurso ao jogo: coaches. Essas cartas permitem que os jogadores comprem cartas com mais facilidade e ainda fornecem bônus adicionais.

Nikolle, uma menina sortuda, no último pacotinho de cartas que abriu, tirou uma das cartas coach mais raras e poderosas que existem atualmente: o lendário viniciusrpb dourado. Ao jogar essa carta, seu bônus permite que o jogador compre uma carta de seu baralho juntamente com a k-ésima carta com poder de habilidade maior ou igual à primeira carta retirada (considerando as cartas na ordem da esquerda para a direita).

O lendário viniciusrpb dourado

Dada a configuração do baralho de Nikolle, para cada carta no baralho, ajude-a a encontrar a carta a ser também comprada ao jogar o viniciusrpb dourado.

Entrada

A primeira linha de entrada contém dois inteiros n e k ($1 \leq n \leq 2 \cdot 10^5$; $1 \leq k \leq 50$) — o número n de cartas na pilha de Nikolle e o parâmetro k.

A segunda linha de entrada contém n inteiros $a_1, \ldots, a_n$ ($1 \leq a_i \leq 10^9$) — a_i é o poder de habilidade da i-ésima carta da esquerda para a direita.

Saída

Imprima n inteiros $b_1, \ldots, b_n$ — o k-ésimo menor índice b_i que é maior que i e que satisfaz $a_{b_i} \geq a_i$, se existir. Caso contrário b_i deve ser 0.

Exemplo

Entrada	Saída
7 1	2 5 5 5 0 7 0
4 6 5 2 9 3 3	
7 2	3 0 0 6 0 0 0
4 6 5 2 9 3 3	
15 3	4 0 9 0 0 12 12 11 0 15 14 0 0 0 0
16 23 18 21 24 8 4 3 20 12 3 17 6 18 13	

Problema M – Mineirinho

Limite de tempo: 6s
Limite de memória: 256MB

Autor: Daniel Saad Nogueira Nunes

Após passar um tempo na Alemanha, regado apenas a cerveja e salsichas, o mineirinho começou a ter fortes crises de abstinência da comida mineira. A falta da gordura do torresmo e do tempero do feijão tropeiro o estava deixando maluco. O mineirinho ficou tão alucinado que começou a ver triângulos em tudo, em uma possível alusão à bandeira do seu estado natal.

Ele então resolveu ir ao Dr. Dreieckmann, médico especialista em alucinações triangulares. O médico disse que essas alucinações só poderiam ser tratadas caso ele fosse capaz de, dentre todos os triângulos que ele visse, encontrar aquele que possuísse o menor perímetro.

Ajude o mineirinho a encontrar o triângulo de menor perímetro antes que ele enlouqueça de vez!

Entrada

A primeira linha da entrada possui um inteiro n, indicando o número de pontos. Cada uma das próximas n linhas possui um par de inteiros x_i e y_i ($-10^9 \leq x_i, y_i \leq 10^9$), indicando as coordenadas do i-ésimo ponto no qual os triângulos poderiam ser formados na visão do mineirinho.

Restrições:

- $3 \leq n \leq 2 \cdot 10^5$;
- $-10^9 \leq x_i, y_i \leq 10^9$, $1 \leq i \leq n$;
- Os pontos são distintos.

Saída

Imprima o menor perímetro dentre todos os triângulos que podem ser formados com os pontos dados. Triângulos degenerados, isto é, cuja área é 0, também são considerados triângulos nesse problema. Se sua resposta é y e a resposta do juiz é x, ela será considerada correta se $\frac{|x-y|}{\max(1,x)} \leq 10^{-5}$

Exemplo

Entrada	Saída
3 2 0 0 0 1 0	4
4 2 1 0 1 -3 0 0 0	5.2360679774997898
3 0 0 0 1 1 0	3.4142135623730949

Notas

No primeiro exemplo, o triângulo, que é degenerado, possui perímetro igual a 4. No segundo exemplo, o mineirinho consegue ver quatro triângulos diferentes, mas o de menor perímetro é aquele formado pelos pontos $(0, 0)$, $(2, 1)$ e $(0, 1)$, com perímetro total igual a $3+\sqrt{5}$. No terceiro exemplo, o único triângulo que o mineirinho consegue ver possui perímetro $2+\sqrt{2}$.

Problema N – Netzwerk

Limite de tempo: 5.5s
Limite de memória: 256MB

Autor: Arthur Botelho

O agente especial **viniciusrpb** está coordenando uma operação secreta da agência de inteligência alemã: a Operação *Netzwerk* (Rede). A operação envolve o transporte de K cargas valiosas por cidades brasileiras a partir de viagens aéreas clandestinas. As cidades são numeradas de 1 a N, e cada uma viagem clandestina é caracterizada por dois números A e B, indicando que pode ser feita uma viagem partindo da cidade A e chegando na cidade B.

Inicialmente, todas as K cargas valiosas estão localizadas em cidades distintas. Essas cargas devem ser transportadas, por meio de sequências de viagens clandestinas, para outras K cidades de destino, também todas distintas. Por motivos de segurança, é preciso que cada cidade de destino receba apenas uma carga valiosa ao final da operação. É permitido que, ao longo da operação, mais de uma carga passe por uma cidade destino, porém apenas uma pode terminar na cidade para ser recebida pelos alemães!

viniciusrpb é bastante cauteloso. Ao planejar o transporte para que todas as cargas cheguem em alguma cidade de destino, ele irá tentar minimizar o risco da operação ser descoberta, que é o máximo entre o risco local de todas as cidades de 1 a N. O risco local de uma cidade é dado pela quantidade de cargas valiosas que passaram por ela ao longo de toda a operação. Por exemplo:

Considere que há 5 cidades numeradas de 1 a 5, e que existem as viagens de 1 para 3, de 2 para 3, de 3 para 4 e de 3 para 5. Supondo que as cargas estão nas cidades 1 e 2 e que as cidades destino são 4 e 5, as cargas poderiam ser transportadas a partir de sequências de viagens que passam pelas cidades (1, 3, 4) e (2, 3, 5). Dessa forma, os riscos locais das cidades de 1 a 5 seriam, respectivamente: 1, 1, 2, 1, e 1, e o risco da operação ser descoberta seria o maior desses valores (2).

Para ajudar **viniciusrpb** no planejamento da operação, informe a ele o menor risco possível da operação ser descoberta, considerando todas as forma possíveis de realizar os transportes das cargas. Caso não seja possível transportar todas as cargas para cidades de destino de forma que cada cidade de destino receba exatamente uma carga, **viniciusrpb** também deve ser informado!

Entrada

A primeira linha da entrada contém dois inteiros N, M, representando a quantidade de cidades e a quantidade de viagens aéreas clandestinas, de forma que $2 \leq N \leq 5 \cdot 10^3$ e $1 \leq M \leq 10^4$.

A próxima linha da entrada contém N inteiros t_i ($1 \leq i \leq N$), que descreve se a cidade i possui uma carga valiosa ($t_i = 1$), se é uma cidade destino ($t_i = 2$) ou se não atende a nenhuma dessas condições ($t_i = 0$). É garantido que a quantidade de valores $t_i = 1$ será igual à quantidade de valores $t_i = 2$, essa quantidade sendo o valor K: o número de cargas valiosas e de cidades destino. Também é garantido que haverá pelo menos um $t_i = 1$.

As próximas M linhas da entrada contêm dois inteiros A_i e B_i cada, representando que existe uma viagem clandestina partindo da cidade A_i e chegando na cidade B_i. É garantido que $1 \leq A_i, B_i \leq N$ e que $A_i \neq B_i$.

Saída

Imprima uma única linha contendo um único inteiro (com base nas informações da entrada): o menor risco possível para a operação, se for possível transportar todas as cargas para cidades de destino de forma que cada cidade de destino receba exatamente uma carga, ou -1, caso contrário.

Exemplo

Entrada	Saída
5 4 1 1 0 2 2 1 3 2 3 3 4 3 5	2
6 6 1 1 0 0 2 2 1 3 2 4 3 4 4 3 3 5 4 6	1
4 2 1 1 2 2 1 3 1 4	-1

Notas

O primeiro caso de exemplo é explicado no enunciado.

No segundo caso de exemplo, poderia-se transportar as cargas por viagens passando por $(1, 3, 4, 6)$ e $(2, 4, 3, 5)$, o que resultaria em um risco igual a 2 devido às cidades 3 e 4. Para minimizar o risco, deve-se fazer o transporte passando por $(1, 3, 5)$ e $(2, 4, 6)$.

No terceiro caso, é impossível transportar a carga da cidade 2 para alguma cidade destino (por isso, deve-se imprimir -1).