Seletiva UnB 2026

Caderno de Problemas

27 de junho de 2026

(Este caderno contém 15 problemas)

Comissão Organizadora:
Alberto Tavares Duarte Neto
Daniel Porto
Daniel Saad Nogueira Nunes
Edson Alves da Costa Júnior
Guilherme Novaes Ramos
Jeremias Moreira Gomes
José Marcos Leite
Pedro de Carvalho Gallo Pereira
Ruan Petrus Alves Leite

Universidade de Brasília - UnB - Campus Darcy Ribeiro

Orientações

- 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)`.
- É sabido que em alguns casos de problemas com entradas muito grandes, os objetos `cin` podem ser lentos, por conta da sincronização de buffers da biblioteca `stdio`.
- Caso se deseje utilizar `cin` e `cout`, um jeito de se contornar este problema é executando-se o comando `ios_base::sync_with_stdio(0)`, no início de sua função `main`.
- Note que, neste caso, o uso de `scanf` e `printf` no mesmo programa é contra-indicado, uma vez que, com buffers separados, comportamentos inadequados podem ocorrer.

Java

- Não declare "package" no seu programa Java.
- Note que a convenção para o nome do arquivo fonte deve ser obedecida, o que significa que o nome de sua classe pública deve ser uma letra maiúscula (A, B, C, etc).
- Comando para executar uma solução Java:
  `java -Xms1024m -Xmx1024m -Xss100m codigo_do_problema`

Kotlin

- Não declare "package" no seu programa Kotlin.
- Note que a convenção para o nome do arquivo de solução deve ser obedecida, o que significa que seu arquivo-fonte deve ser nomeado (A.kt, B.kt, C.kt, etc).
- Comando para executar uma solução Kotlin:
  `java -Xms1024m -Xmx1024m -Xss100m codigo_do_problemaKt`

Python

- Note que apenas Python 3 é suportado.
- As submissões em Python terão sua sintaxe verificada; programas que não passarem nessa verificação receberão o veredito "Compilation Error".

Problema A – Adelaide

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

Autor: Guilherme Ramos

Verticalmente desafiada, Adelaide andava na rua à noite, totalmente só. Vez ou outra via coisas, em bancas de jornal, devaneios de seu saudoso Paraguai... Tímida, mas sonhadora, sempre imaginava como seriam as conversas com o basquetebolista lusitano que admirava.

Ajude Adelaide a desenvolver os diálogos!

Entrada

A entrada consiste na tentativa de Adelaide de engajar o gajo em uma conversa significativa. Uma linha com pelo menos 10 caracteres e não mais de 100.

Saída

Para cada tópico proposto, apresente a resposta padrão do portuga e expresse a ira de Adelaide pela pergunta insípida, conforme os exemplos.

Exemplo

entrada 1

```
soy adelaide, de paraguay. vamos-nos a un botequito para asar un jabali?
```

saída 1

```
soy adelaide, de paraguay. vamos-nos a un botequito para asar un jabali?
para que?
paraguay!
```

entrada 2

```
mi nombre es adelaide, vengo de paraguay. estoy aqui para amarte!
```

saída 2

```
mi nombre es adelaide, vengo de paraguay. estoy aqui para amarte!
para que?
paraguay!
```

entrada 3

```
soy adelaide, de paraguay. parabens a mi portunol?
```

saída 3

```
soy adelaide, de paraguay. parabens a mi portunol?
para que?
paraguay!
```

Notas

Adelaide é uma ótima pessoa, mas excessivamente patriota. Já o “atleta” ibérico é meio desatento...

Problema B – Buraco de Rato

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

Autor: Alberto Neto

Eduardo e Tyzz estão assistindo ao jogo do Brasil da Copa do Mundo! Porém, ratos começam a roer os cabos da televisão. Eles então pedem a ajuda de Gabriel Sales, o Expulsador de Ratos, para combater este problema.

Existem n ratos na casa, e eles serão expulsos para $n+1$ buracos de modo que exatamente n buracos vão conter exatamente 1 rato. As posições dos ratos e buracos estão dispostas em uma reta, e o tempo necessário para levar um rato na posição r para um buraco na posição b é $|r-b|$.

Considerando que Gabriel expulsa os ratos um a um, determine a menor quantidade de tempo possível para que todos os ratos sejam expulsos e Eduardo e Tyzz possam aproveitar o jogo sem interrupções.

Entrada

A primeira linha de entrada contém um único inteiro n ($1 \leq n \leq 2 \cdot 10^5$) — o número de ratos.

A segunda linha de entrada contém n inteiros $r_1, \ldots, r_n$ ($0 \leq r_i \leq 10^9$; $r_i < r_{i+1}$) — a posição dos ratos.

A terceira linha de entrada contém $n+1$ inteiros $b_1, \ldots, b_n, b_{n+1}$ ($0 \leq b_i \leq 10^9$; $b_i < b_{i+1}$) — a posição dos buracos.

Saída

Imprima um único inteiro — o tempo mínimo necessário para que todos os ratos estejam em um buraco.

Exemplo

Entrada	Saída
2 7 12 4 11 15	4
2 7 18 4 11 15	6

Notas

No primeiro caso de teste, o rato da posição 7 vai para o buraco da posição 4, e o rato da posição 12 vai para o buraco da posição 11. O tempo necessário para expulsar os ratos é $|7-4| + |12-11| = 4$.

Problema C – Critical Hit!

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

Autor: Edson Alves

No RPG "Ultimate Tatics" dois jogadores, que assumem o papel de generais, se enfrentam em batalhas épicas. Em uma homenagem a um clássico do gênero, o jogo disponibiliza, dentre as unidades que podem ser selecionadas para compor o exército do jogador, a classe favorita dos estudantes de ciências exatas: o Matemático!

O Matemático lança suas magias sem gastar mana: o jogador escolhe um número D maior do que um e são afetadas todas as unidades adversárias cujo número de pontos de defesa é divisível por D. Se D for um quadrado perfeito, então o acerto é crítico, triplicando o dano!

Cansado de perder para o professor Saad, o professor Costa precisa de ajuda para montar um exército que tenha alguma chance de vencer os poderosos matemáticos do exército do professor Saad. Dados os pontos de defesa d_i das N unidades que podem ser selecionadas, determine quais delas são imunes a danos críticos.

Entrada

A primeira linha da encontra contém o inteiro N ($1 \leq N \leq 2 \times 10^5$), que indica o número de unidades que podem ser selecionadas pelo professor Costa para montar o seu exército.

A segunda linha contém N inteiros d_i ($1 \leq d_i \leq 10^7, 1 \leq i \leq N$), separados por um espaço em branco, em que d_i é o número de pontos de defesa da unidade i.

Saída

Imprima, em uma linha, o número M de unidades, dentre as N indicadas na entrada, que são imunes aos danos críticos dos matemáticos.

Na linha seguinte imprima M inteiros j, separados por um espaço em branco, indicando que a j-ésima unidade é imune aos danos críticos.

Exemplo

Entrada	Saída
3 12 15 18	1 2
5 100 200 300 400 500	0
2 61 77	2 1 2

Notas

No primeiro caso, a unidade 1 pode ser afetada por um dano crítico se $D = 4$; já a unidade 3 sofre dano crítico com $D = 9$. Dentre as três, apenas a unidade 2 é imune aos danos críticos.

No segundo caso, todas as unidades podem ser afetadas por danos críticos caso o professor Saad escolha $D = 100$.

Problema D – Descompressão FoR

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

Autor: Daniel Saad Nogueira Nunes

A técnica FoR (Frame of Reference) permite comprimir inteiros pequenos em poucos *bits* e funciona basicamente da seguinte forma: dado um bloco de inteiros, escolhe-se um dentre eles como referência, normalmente o menor, e subtrai-se esse número de todos os outros elementos do bloco. Os inteiros resultantes são menores do que os originais e podem ser representados com menos *bits*. Cada bloco é anexado ao inteiro de referência e ao número de *bits* necessários para representar os valores comprimidos daquele bloco.

Por exemplo, se o bloco de inteiros for $\langle 255, 250, 253, 254 \rangle$, o valor de referência é 250. O bloco resultante após sua aplicação é $\langle 5, 0, 3, 4 \rangle$, de modo que cada número pode ser representado com 3 *bits*, em vez dos 8 necessários originalmente.

Para obter a sequência original, basta adicionar o inteiro de referência após descomprimir cada inteiro do bloco. No exemplo acima, ao adicionar 250 a cada inteiro do bloco $\langle 5, 0, 3, 4 \rangle$, obtemos novamente o bloco original $\langle 255, 250, 253, 254 \rangle$.

Neste problema, você deve implementar a descompressão FoR para inteiros de 8 *bits*, dada a quantidade total de números comprimidos e a de inteiros por bloco. Cada bloco inicia com um cabeçalho de 11 *bits*, sendo os primeiros 8 *bits* para o inteiro de referência e os últimos 3 *bits* para a quantidade de *bits* necessários para representar os inteiros comprimidos. Além disso, considere que cada inteiro é compactado a partir dos *bits* menos significativos (veja o exemplo ao final).

Entrada

A primeira linha da entrada contém dois inteiros n e b ($1 \leq n \leq 2^{20}$, $1 \leq b \leq 2^{10}$), representando respectivamente a quantidade total de inteiros comprimidos e a quantidade de inteiros por bloco.

A segunda linha possui uma string hexadecimal com a codificação dos n inteiros em FoR. Os inteiros comprimidos estão no intervalo de 0 a 255, e a string hexadecimal possui apenas os caracteres de 0 a 9 e de a a f. Cada *byte* da informação comprimida é representado com dois caracteres hexadecimais e, possivelmente, o último *byte* pode ter *bits* não utilizados (*padding*).

Saída

Imprima a sequência de inteiros descomprimidos, um inteiro por linha.

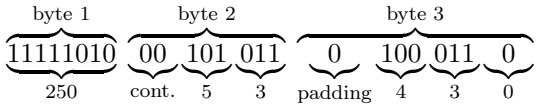
Exemplo

Entrada	Saída
4 4 fa2b46	255 250 253 254
12 4 fa2b46fd15a3fe8a11	255 250 253 254 255 250 253 254 255 250 253 254

Notas

Esse problema é custoso em operações de entrada e saída, então é recomendado o uso de técnicas rápidas de leitura e escrita.

O primeiro exemplo reflete o descrito no enunciado. Há um único bloco de tamanho 4 e os dados comprimidos em hexadecimal são fa2b46, que representam o padrão binário



O primeiro *byte* representa o inteiro de referência 250. Os três primeiros *bits* do segundo *byte* correspondem ao inteiro 3, que é a quantidade de *bits* necessários para representar os inteiros comprimidos. Os próximos 4 blocos de três *bits* representam os inteiros comprimidos 5, 0, 3 e 4. Note que o inteiro 0 tem dois bits codificados nos *bits* mais significativos do segundo *byte* e o terceiro *bit* codificado no início do último *byte*. O último *bit* do último *byte* é um preenchimento, que não é utilizado.

O segundo exemplo é uma concatenação de três cópias do primeiro exemplo, representando a sequência de inteiros $\langle 255, 250, 253, 254, 255, 250, 253, 254, 255, 250, 253, 254 \rangle$.

Problema E – Exemplos e Contraexemplos

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

Autor: Edson Alves

Preparando mais uma de suas excelentes aulas, o professor Neto escolheu, como tema do próximo encontro, a seguinte proposição: "O produto de quatro inteiros positivos consecutivos é sempre o antecessor de um quadrado perfeito".

Sabendo que a recíproca não é verdadeira, isto é, nem todo quadrado perfeito é sucessor do produto de quatro inteiros positivos consecutivos, ele quer separar alguns exemplos e contraexemplos desta recíproca, para ilustar o fato a seus estudantes. Como ele tem coisas mais importantes para fazer, como procurar uma demonstração para a hipótese de Riemann, ele deixou esta tarefa para você: dado o valor de um inteiro N, determine se N^2 é ou não o sucessor do produto de quatro inteiros positivos consecutivos.

Entrada

A entrada consiste em uma única linha, a qual contém o valor do inteiro N ($1 \leq N \leq 10^{18}$).

Saída

Imprima, em uma linha, o inteiro positivo m tal que $m(m+1)(m+2)(m+3) = N^2 - 1$. Caso não exista tal inteiro, imprima o valor -1.

Exemplo

Entrada	Saída
5	1
10	-1
15241754305	123456

Notas

No primeiro caso,

$$1 \times 2 \times 3 \times 4 = 24 = 25 - 1 = 5^2 - 1$$

No segundo caso, não existe uma sequência de quatro inteiros positivos consecutivos tais que o produto seja o antecessor de $10^2 = 100$.

Problema F – Figurinhas da Copa

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

Autor: Daniel Saad Nogueira Nunes

O álbum de figurinhas da Copa do Mundo de 2026 virou uma febre entre os colecionadores. Existem figurinhas de jogadores, de estádios, de mascotes, de bandeiras, de escudos, e por aí vai. Algumas figurinhas são comuns e outras são bem raras. Em especial, existe uma super rara, a figurinha do jogador da seleção Brasileira, Guillermo Ramos, primo do jogador espanhol Sérgio Ramos, mas que foi naturalizado brasileiro. No momento, existe apenas uma figurinha de G. Ramos no mundo.

Como você é um grande fã de futebol e das figurinhas, e mais ainda de dinheiro, quer especular e ganhar uma grana com a figurinha do Guillermo. Em n dias, você conhecerá o preço de compra ou venda da figurinha do Guillermo. Você pode comprar a figurinha em um dia e vendê-la em outro dia posterior, ou seja, você não pode vender a figurinha antes de comprá-la. Além disso, você pode repetir o processo de compra e venda sempre que quiser. Após os n dias, a editora fabricará centenas de milhares de figurinhas do Guillermo, e o preço dela cairá para zero.

Qual é o maior lucro que você pode obter com as compras e vendas da figurinha do Guillermo, considerando que você não a possui inicialmente?

Entrada

A primeira linha da entrada contém um inteiro n ($1 \leq N \leq 10^6$), o número de dias até a editora fabricar figurinhas do G. Ramos. A segunda linha contém n inteiros $p_1, p_2, \ldots, p_n$ ($0 \leq p_i \leq 10^9$, $1 \leq i \leq n$), em que p_i é o preço de compra ou venda da figurinha de G. Ramos no i-ésimo dia.

Saída

Imprima uma linha com o maior lucro que você pode obter com as compras e vendas da figurinha do Guillermo.

Exemplo

Entrada	Saída
5 1 2 3 4 5	4
4 10 5 3 1	0
6 1 5 10 2 8 5	15

Notas

No primeiro exemplo, você deve comprar a figurinha no primeiro dia por 1 e vendê-la no quinto dia por 5, obtendo um lucro de 4. No segundo exemplo, é melhor não brincar com isso, senão você levará prejuízo. No terceiro exemplo, você pode comprar a figurinha no primeiro dia por 1, vendê-la no terceiro dia por 10, comprá-la novamente no quarto dia por 2 e vendê-la no quinto dia por 8, obtendo um lucro total de $9 + 6 = 15$.

Problema G – Genomas Revertidos

Limite de tempo: 6s
Limite de memória: 1024MB

Autor: Daniel Saad Nogueira Nunes

Além de mutações, genomas podem sofrer reversões, isto é, blocos de genes podem ter a sua orientação completamente invertida. Esse é um aspecto crucial na genômica comparativa. Estimar quantas reversões de genoma foram realizadas para sair de um genoma A de um organismo para o genoma B de outro organismo pode ser um indicativo da distância evolutiva desses dois organismos.

Modelando matematicamente, podemos simplificar a visão de um genoma como uma sequência de genes, em que cada gene é um inteiro. Ou seja, um genoma pode ser visto como uma permutação $\pi = \langle p_1, \ldots, p_i, p_{i+1}, \ldots, p_{j-1}, p_j, \ldots, p_n \rangle$ de $\{1, \ldots n\}$. Uma reversão de π no intervalo $[i, j]$ nos fornece a permutação $\langle p_1, \ldots, p_{i-1}, p_j, p_{j-1}, \ldots, p_{i+1}, p_i, p_{j+1}, \ldots, p_n \rangle$.

Calcular a distância evolutiva de dois genomas π_A e π_B conforme a métrica de reversões é calcular o número mínimoo de reversões necessárias para transformar o genoma A no genoma B. Esse problema ainda pode ser reduzido ao problema de calcular a distância de reversões de uma permutação $\pi_{A'}$, que é π_A rearranjada, para a permutação identidade $\{1, \ldots, n\}$.

Calcule a distância de reversões de uma permutação π qualquer para a permutação identidade.

Entrada

A primeira linha da entrada possui um inteiro n ($1 \leq n \leq 10$). A segunda linha da entrada possui n inteiros, separados por um espaço, descrevendo uma permutação π de $\{1, \ldots, n\}$.

Saída

Imprima a distância de reversões de π para a permutação identidade.

Exemplo

Entrada	Saída
4 1 2 3 4	0
4 4 3 2 1	1
4 2 1 3 4	1
4 2 1 4 3	2
4 3 1 2 4	2

Notas

No primeiro exemplo, a permutação π é a permutação identidade. Não é necessário fazer nenhuma reversão. A resposta é 0.

No segundo exemplo, a permutação π é $\langle 4, 3, 2, 1 \rangle$. Revertendo o intervalo $[1, 4]$ obtemos a permutação identidade. A resposta é 1.

No terceiro exemplo, a permutação π é $\langle 2, 1, 3, 4 \rangle$. Revertendo o intervalo $[1, 2]$, chegamos na permutação identidade. A resposta é 1.

No quarto exemplo, a permutação π é $\langle 2, 1, 4, 3 \rangle$. Chegamos na permutação identidade ao aplicar as seguintes reversões: $[1, 2]$ e $[3, 4]$. A resposta é 2.

No quinto exemplo, a permutação π é $\langle 3, 1, 2, 4 \rangle$. Revertendo o intervalo $[1, 2]$ seguido do intervalo $[2, 3]$, obtemos a permutação identidade. A resposta é 2.

Problema H – Horda de Gols

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

Autor: Jeremias Moreira Gomes

A Copa do Mundo de 2026 será realizada nos Estados Unidos, e a União Stadunidense de Serviços de Futebol (USSF) é a responsável pela infraestrutura da competição no país. O único problema é que os americanos não conhecem muito bem as regras do futebol e, ao lerem sobre fase de grupos, eles entenderam errado e acabaram construindo estádios com apenas um campo, onde todas as seleções jogam ao mesmo tempo.

Agora que não tem mais como voltar atrás, eles decidiram continuar com a competição e utilizar as seguintes regras: cada vez que uma equipe faz um gol, ela recebe um ponto; porém, se a equipe marcar um gol contra, ela perde um ponto. Como só há um juiz, cada gol é registrado individualmente, nunca ocorrendo dois registros no mesmo instante de tempo (o VAR garante que nenhum gol seja perdido). Toda partida há sempre um vencedor, e a equipe vitoriosa é aquela que tiver a maior pontuação ao final da partida. Caso duas ou mais equipes terminem com a mesma pontuação máxima, será declarada vencedora aquela cuja última alteração de pontuação tenha ocorrido primeiro durante a partida.

Como as regras podem ficar confusas para os torcedores e outros países, a USSF pediu a sua ajuda para escrever um programa que, dados os gols marcados durante uma partida, determine qual equipe será a campeã.

Entrada

A entrada contém um único caso de teste. A primeira linha contém um inteiro E ($1 \leq E \leq 6000$), que representa a quantidade de eventos ocorridos durante a partida. Cada uma das próximas E linhas contém o nome de uma equipe S (uma string composta por caracteres alfabéticos minúsculos, com $1 \leq |S| \leq 50$), um instante no formato `MM:SS` e um inteiro G ($G \in \{-1, 1\}$), indicando que naquele instante a equipe marcou um gol (se $G = 1$) ou um gol contra (se $G = -1$).

Saída

A saída deve conter uma única linha com o nome da equipe vencedora da partida.

Exemplo

Entrada	Saída
6 brasil 15:00 1 brasil 45:00 1 marrocos 80:00 1 escocia 60:00 1 gana 37:00 1 escocia 30:00 1	brasil
10 canada 02:00 1 suica 55:00 1 suica 60:00 1 suica 30:00 1 catar 32:00 1 canada 26:00 1 suica 25:00 -1 canada 45:00 1 catar 68:00 1 catar 92:00 1	canada

Problema I – Incrível metrô

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

Autor: Alberto Neto

Pedro Gallo foi contratado para resolver um grande problema do metrô de Brasília. Foram construídas n estações, com cada uma possuindo **no máximo** um trilho até outra estação. No planejamento da cidade foram levantados m pares de estações (u_i, v_i) tais que deve existir um caminho de u_i até v_i, e o metrô foi feito para atender a essa demanda.

O problema é que entre as estações com trilhos foi construído apenas uma linha de trem, e não um par, de forma que os trens andam apenas em uma direção. Pedro Gallo, porém, se pergunta: é possível direcionar os trilhos para que todos os caminhos sejam satisfeitos?

Ajude Pedro Gallo e determine se é possível direcionar os trilhos e resolver o problema. Se houver solução, diga qual deve ser a direção de cada trilho.

Entrada

A primeira linha de entrada contém um inteiro n ($2 \leq n \leq 2 \cdot 10^5$) — o número de estações de trem.

A segunda linha de entrada contém n inteiros $a_1, \ldots, a_n$ ($1 \leq a_i \leq n$) — representando que há um caminho direcionado da estação i para a estação a_i se $i \neq a_i$, ou que não há um caminho caso contrário.

A terceira linha de entrada contém um inteiro m ($1 \leq m \leq 2 \cdot 10^5$) — o número de caminhos a serem satisfeitos.

As próximas m linhas de entrada contém dois inteiros u_i, v_i ($1 \leq u_i, v_i \leq n$) — representando que deve existir um caminho de u_i até v_i.

Saída

Se não há resposta, imprima -1.

Caso contrário, imprima uma linha contendo n inteiros $b_1, \ldots, b_n$ — com $b_i = 1$ se o caminho de i para a_i deve ser invertido, e $b_i = 0$ caso contrário. Se $a_i = i$, imprima 0 ou 1.

Se houver mais de uma resposta correta, imprima qualquer uma.

Exemplo

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

Problema J – Juan e o Dicionário

Limite de tempo: 1s
Limite de memória: 512MB

Autor: Ruan Petrus

Juan está construindo um dicionário para um jogo de palavras.

Nesse dicionário, uma regra deve ser sempre respeitada: sempre que uma palavra estiver cadastrada, todos os seus prefixos não vazios também devem estar cadastrados. Por exemplo, se a palavra `casa` pertence ao dicionário, então as palavras `c`, `ca` e `cas` também devem pertencer.

Você recebe uma string S de tamanho N, indexada de 1 a N.

Inicialmente, o dicionário está vazio. Em seguida, são realizadas Q operações. Na i-ésima operação, Juan escolhe a substring $S[L_i..R_i]$ e deseja adicioná-la ao dicionário.

Antes de adicioná-la, ele precisa inserir todas as palavras necessárias para que a regra do dicionário continue sendo satisfeita.

Após cada operação, determine quantas palavras existem no menor dicionário possível que contém todas as palavras escolhidas até aquele momento.

Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 2 \cdot 10^5$).

A segunda linha contém a string S, composta apenas por letras minúsculas do alfabeto.

A terceira linha contém um inteiro Q ($1 \leq Q \leq 2 \cdot 10^5$).

Cada uma das próximas Q linhas contém dois inteiros L_i e R_i ($1 \leq L_i \leq R_i \leq N$), indicando a substring adicionada ao dicionário.

Saída

Para cada uma das Q operações, imprima uma linha contendo um único inteiro: o número de palavras no menor dicionário que satisfaz a regra e contém todas as palavras adicionadas até aquele momento.

Exemplo

Entrada	Saída
5 ababa 3 1 3 2 4 3 5	3 6 6
2 aa 3 1 1 2 2 1 2	1 1 2

Notas

As notas a seguir referem-se ao primeiro caso de teste.

Na primeira operação, Juan adiciona a palavra `aba`. Para que o dicionário continue válido, também é necessário incluir todos os seus prefixos:

`{a, ab, aba}`

Assim, o dicionário possui 3 palavras.

Na segunda operação, a palavra adicionada é `bab`. Seus prefixos são:

`{b, ba, bab}`

Como nenhuma dessas palavras estava presente anteriormente, o menor dicionário passa a ser:

`{a, ab, aba, b, ba, bab}`

totalizando 6 palavras.

Na terceira operação, a palavra adicionada é novamente `aba`, que já pertence ao dicionário. Portanto, nenhuma nova palavra precisa ser inserida e a resposta permanece 6.

Problema K – Kamzabek, o Kazakh

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

Autor: Alberto Neto

Hoje é dia de Seletiva da UnB! Na madrugada antes da prova, Nson percebe que o problema K foi roubado pela Unicamp como forma de sabotagem. Como na China é dia, ele liga para Tiagodfs, que aciona seu amigo Kamzabek. Eles entregam a Nson o seguinte problema.

Você recebe duas permutações l e r de tamanho n, e um inteiro k.

Uma permutação p é chamada *chinesa* se as seguintes condições valem:

- $l \leq p \leq r$ na ordem lexicográfica;
- o número de inversões na permutação p é igual a k.

Sua tarefa é calcular o número de inversões chinesas módulo 998244353.

Entrada

A primeira linha contém dois inteiros n e k ($1 \leq n \leq 10^6$, $0 \leq k \leq 5000$).
A segunda linha contém n inteiros $l_1, l_2, \ldots, l_n$ — os elementos da permutação l.
A terceira linha contém n inteiros $r_1, r_2, \ldots, r_n$ — os elementos da permutação r.
É garantido que l e r são permutações de tamanho n, e que $l \leq r$ em ordem lexicográfica.

Saída

Imprima um único inteiro — o número de permutações p tais que $l \leq p \leq r$ em ordem lexicográfica e o número de inversões em p seja igual a k, módulo 998244353.

Exemplo

Entrada	Saída
4 3 1 2 3 4 4 3 2 1	6
6 5 2 1 3 4 5 6 4 2 6 1 5 3	43

Notas

Uma permutação de tamanho n é um vetor consistindo de n inteiros distintos de 1 até n.

Uma permutação a é lexicograficamente menor que uma permutação b se existe uma posição i tal que:

- $a_j = b_j$ para todo $1 \leq j < i$;
- $a_i < b_i$.

Uma inversão em uma permutação p é um par de índices (i, j) tal que $1 \leq i < j \leq n$ e $p_i > p_j$.

Problema L – Lobotomia Mental

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

Autor: Jeremias Moreira Gomes

Daniel está lecionando na turma de APC (Algoritmos e Programação para Crianças) e já não aguenta mais os alunos cantando essa música o dia todo:

```
Vinte mais vinte mais vinte mais sete,
Aí é muito fácil, professor, é six seven.
```

Tentando entrar no tópico da aula, enquanto essa música chiclete não saía da cabeça, ele fez o seguinte comentário durante a aula: "Sabia que qualquer número inteiro positivo pode ser representado utilizando apenas dois símbolos?". Automaticamente, a turma toda respondeu: "Six Seven". Não aguentando mais e se entregando à loucura, Daniel riu e escreveu o seguinte no quadro:

```
1 -> six
2 -> six seven
3 -> six six
4 -> six seven seven
5 -> six seven six
6 -> six six seven
7 -> six six six
8 -> six seven seven seven
...
```

As crianças ficaram fascinadas com a ideia, e começaram a escrever todos os números em "six seven". Daniel foi ainda além: pediu para que os alunos ordenassem uma lista de números inteiros positivos pela sua cardinalidade de "six" de cada número e, em caso de empate, o menor número deve vir primeiro.

Agora que Daniel precisa corrigir a tarefa. Ele pediu a sua ajuda para escrever um programa que realize essa ordenação.

Entrada

A entrada possui um único caso de teste. A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 10^5$), que indica a quantidade de números que Daniel passou para os alunos. A segunda linha contém N inteiros X ($1 \leq X \leq 10^8$), separados por um espaço, que são os números que devem ser ordenados.

Saída

A saída deve conter uma única linha com os N números ordenados, segundo a cardinalidade de "six" de cada número.

Exemplo

Entrada	Saída
10 4 1 8 3 9 2 3 8 7 5	1 2 4 8 8 3 3 5 9 7
11 18 4 2 5 18 8 10 5 4 16 9	2 4 4 8 16 5 5 9 10 18 18

Problema M – Maior

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

Autor: Edson Alves

Seja n um número positivo,

$$D(n) = \{\ x \mid x \text{ divide } \ n\ \}$$

e

$$I = \{\ x \mid x \text{ é ímpar } \ \}$$

Dado o valor de n, determine o maior elemento da interseção entre ambos conjunto, isto é, determine o inteiro m tal que

$$m = \max D(n) \cup I$$

Por exemplo, para $n = 20$ temos que

$$D(20) = \{\ -20, -10, -5, -4, -1, 1, 2, 4, 5, 10, 20\ \}$$

e

$$I = \{\ \ldots, -5, -3, -1, 1, 3, 5, \ldots\ \},$$

de modo que $m = \max D(20) \cup I = 5$.

Entrada

A entrada contém uma única linha, com o valor do inteiro N ($1 \leq N \leq 10^{18}$).

Saída

Imprima, em uma linha, o valor de m.

Exemplo

Entrada	Saída
20	5
3	3
395040	12345

Notas

O primeiro caso foi ilustrado no texto do problema.

Problema N – Noite Estrelada

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

Autor: Pedro Pereira

Durante uma viagem ao Alasca, um grupo de aventureiros decidiu procurar um bom lugar para acampar e apreciar a noite estrelada junto da aurora boreal. Para evitar áreas perigosas, como penhascos, lagos congelados e trechos cobertos por neve instável, eles delimitaram uma região segura em formato de polígono.

Depois disso, escolheram um ponto p como posição candidata para montar a fogueira principal do acampamento. Para que essa posição seja considerada segura, os aventureiros precisam conseguir caminhar em linha reta da fogueira até qualquer canto da região demarcada sem sair da área segura.

Mais precisamente, para cada vértice v da área segura, o segmento de reta que liga p a v deve estar totalmente contido na área segura, incluindo sua borda.

Sua tarefa é determinar se a posição candidata p é segura.

Entrada

A primeira linha contém dois inteiros p_x e p_y ($-10^9 \leq p_x, p_y \leq 10^9$) — as coordenadas do ponto p, o qual indica a posição candidata para a fogueira principal.

A segunda linha contém um inteiro n ($3 \leq n \leq 2 \cdot 10^5$) — o número de vértices da área segura.

As próximas n linhas contêm dois inteiros x_i e y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — as coordenadas dos vértices da área segura, em ordem anti-horária.

Saída

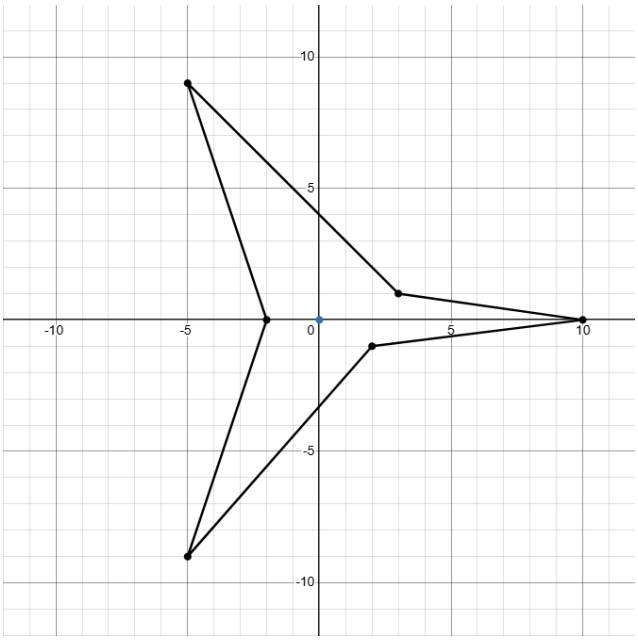
Imprima, em uma linha, a mensagem "`Sim`", se a posição candidata p é segura. Caso contrário, imprima a mensagem "`Nao`".

Exemplo

Entrada	Saída
0 0 4 -1 -1 1 -1 1 1 -1 1	Sim
0 2 5 -2 -2 2 -2 2 2 0 1 -2 2	Nao
0 0 6 10 0 3 1 -5 9 -2 0 -5 -9 2 -1	Sim

Notas

A imagem abaixo ilustra o terceiro caso de teste. O ponto azul representa a posição candidata p para a fogueira.



Nesse caso, a posição candidata é segura: para todo vértice da área segura, o segmento de reta que liga p a esse vértice fica totalmente contido na área segura.

Problema O – Ovos

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

Autor: Edson Alves

Já Kan é um experiente chefe de cozinha, cuja especialidade são receitas baseadas em ovos. Embora sejam receitas simples, que todos conhecem, ele consegue dar a elas um sabor inigualável.

O cardápio oferece, no momento, as três opções abaixo (entre parêntesis a quantidade de ovos necessários para o preparo), onde cada prato serve duas pessoas e é identificado pelo número que antecede sua descrição:

1. Ovos mexidos (2 ovos);
2. Ovos pochê (3 ovos);
3. Omelete (5 ovos).

Você foi contratado para estagiar com o famoso *chef*, e sua primeira tarefa é cuidar da compra diária de ovos. Deve ser adquirida uma quantidade de ovos tal que o estoque passe a conter exatamente a quantidade de ovos necessária para preparar os N pedidos feitos no dia anterior, levando em consideração que no estoque há, antes da compra, O ovos.

Dados os valores de N, O e os identificadores dos pratos pedidos no dia anterior, determine quantos ovos devem ser comprados.

Entrada

A primeira linha da entrada contém dois inteiros N e O ($1 \leq N \leq 200, 0 \leq O \leq 10^3$), separados por um espaço em branco, indicando o número de pedidos feitos no dia anterior e o número de ovos no estoque antes da compra, respectivamente.

A segunda linha contém N inteiros p_i ($1 \leq p_i \leq 3, 1 \leq i \leq N$), separados por um espaço em branco, em que p_i representa o identificador do pedido feito pelo i-ésimo cliente.

Saída

Imprima, em uma linha, a quantidade de ovos que deve ser adquirida para atender o critério apresentado.

Exemplo

Entrada	Saída
3 8 1 2 3	2
5 20 1 2 1 3 1	0

Notas

No primeiro caso, foram gastos 2 ovos com o pedido 1, 3 ovos com o pedido 2 e 5 ovos com o pedido 3, em um total de $2 + 3 + 5 = 10$ ovos. Como já haviam 8 ovos em estoque, devem ser adquiridos apenas mais 2 ovos.

No segundo caso, foram gastos 14 ovos no dia anterior. Como o estoque já tem 20 ovos, não é necessário comprar mais nenhum ovo.