

Equipe: \_\_\_\_\_

# Programação Competitiva - Desafios Caderno de Tarefas

11 de agosto de 2025

A PROVA TEM DURAÇÃO DE 3 HORAS

# Instruções

## LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 15 páginas (não contando a capa e folha de rosto), numeradas de 1 a 15. Verifique se o caderno está completo.
- A prova deve ser feita em equipe.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova.
- É permitida a consulta ao help do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa; em particular, seu programa não deve escrever frases como “Digite o dado de entrada:” ou similares.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas não estão necessariamente ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo .c; soluções na linguagem C++ devem ser arquivos com sufixo .cc ou .cpp; soluções na linguagem Java devem ser arquivos com sufixo .java e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 3 devem ser arquivos com sufixo .py3; e soluções na linguagem Javascript devem ter arquivos com sufixo .js.
- Na linguagem Java, não use o comando package, e note que o nome de sua classe principal deve usar somente letras minúsculas (o mesmo nome do arquivo indicado nas tarefas).
- Para tarefas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada tarefa.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou pen-drive, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
  - em C: scanf, getchar, printf, putchar
  - em C++: as mesmas de C ou os objetos cout e cin
  - em Java: qualquer classe ou função padrão, como por exemplo Scanner, BufferedReader, BufferedWriter e System.out.println
  - em Python: read, readline, readlines, input, print, write
  - em Javascript: scanf, printf
- Procure resolver a tarefa de maneira eficiente. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

## DICAS PARA EVITAR ERROS NO NEPS - C, JAVA E PYTHON

Principais armadilhas:

- Classe principal com nome errado (Java): Use sempre `public class Main` — o Neps exige essa convenção.
- Esperar mais entradas do que o problema fornece: Leia exatamente o que o enunciado pede. Nada além disso.
- Imprimir coisas a mais do que o problema pede: Evite mensagens como 'Digite o valor:' ou quebras de linha extras.
- Contar ou calcular errado por lógica incompleta: Preste atenção na lógica de repetição e nas condições de parada.
- Não fechar o Scanner (Java): Feche com `sc.close()` para manter boas práticas.
- Nome da função principal diferente (C): Use `int main()` e retorne 0 ao final.
- Declarar variáveis que não usa ou esquecer de inicializar: Mantenha o código limpo e seguro.
- Confundir leitura de char e string em C: Use `char nome[100]; scanf("%s", nome);` corretamente.
- Usar float ou double quando o problema pede inteiros: Ajuste os tipos ao que o problema
- Confundir divisão inteira e real (Python e Java): Use `//` no Python e cuidado com `int/int` em Java.
- Usar métodos/libraries não suportados pelo Neps: Evite recursos complexos ou avançados sem necessidade.

## Chocolate

Por lei, na Nlogônia todas as barras de chocolate são quadradas. Anamaria tem uma barra quadrada de chocolate de lado  $L$ , que ela quer compartilhar com alguns colegas da obi. Mas ela é uma boa cidadã e cumpre a lei. Então, ela divide a barra em quatro pedaços quadrados, de lado  $L=L/2$ . Depois, ela repete esse procedimento com cada pedaço gerado, sucessivamente, enquanto o lado for maior do que, ou igual a 2cm. Você deve escrever um programa que, dado o lado  $L$  da barra inicial, em centímetros, determina quantos pedaços haverá ao final do processo.

### Entrada

A entrada consiste de uma linha, com um único inteiro,  $L$ , o número de centímetros do lado do quadrado.

### Saída

Seu programa deve imprimir uma única linha, contendo um único inteiro, igual ao número total de pedaços obtidos pela Anamaria.

### Restrições

- $2 \leq L \leq 10^4$

### Informações sobre a pontuação

- Para um subconjunto dos casos de teste totalizando 30 pontos  $L < 64$ .

### Exemplos:

Entrada	Saída
4	16
Entrada	Saída
9	64
Entrada	Saída
2	4

## Supermercado

Maria está participando de um programa de intercâmbio no reino da Nlogônia. Ela está gostando muito da experiência, e decidiu fazer um churrasco para suas novas amigas da escola. Como não tem muito dinheiro, Maria vai fazer uma pesquisa para comprar carne no supermercado mais barato que encontrar.

No entanto, ela está um pouco confusa para saber qual supermercado tem o menor preço. O dinheiro na Nlogônia é o Bit, abreviado por  $B$ , mas não é esse o problema. O problema é que o costume na Nlogônia é informar o preço de uma maneira diferente do que Maria está acostumada. Os preços são anunciados como "X Bits por Y gramas do produto".

Por exemplo, o preço de um dado produto é anunciado como sendo  $B24,00$  por 250 gramas em um supermercado,  $B16,00$  por 100 gramas em outro supermercado,  $B19,00$  por 120 gramas em outro supermercado, e assim por diante.

Você pode ajudar Maria?

Dados os preços anunciados pelos supermercados no bairro em que Maria mora, determine o menor valor que Maria deve gastar para comprar 1 quilograma (1000 gramas) de carne.

### Entrada

A primeira linha contém um número inteiro  $N$ , o número de supermercados próximos à casa de Maria. Cada uma das  $N$  linhas seguintes indica o preço da carne em um supermercado e contém um número real  $P$  e um número inteiro  $G$ , indicando que  $G$  gramas de carne custam  $P$  Bits.

### Saída

Seu programa deve produzir uma única linha, contendo uma única linha, com apenas um número real, o menor preço para comprar 1 quilograma de carne. O resultado deve ser escrito com exatamente dois dígitos após o ponto decimal.

### Restrições

- $1 \leq N \leq 100$
- $0 < P \leq 1000.00$ , representado com dois dígitos após o ponto decimal.
- $1 \leq G \leq 1000$

### Exemplos:

Entrada	Saída
3	20.00
3.0 100	
2.0 100	
5.0 100	

---

Entrada	Saída
4	190.00
100.00 500	
190.00 1000	
200.00 900	
110.00 550	
Entrada	Saída
5	58.56
46.50 794	
25.72 130	
66.00 800	
22.45 110	
38.99 453	

## Corrida

A Federação de Corridas de Charrete (FCC) organiza todo ano a Subida Brigitte Cardoso (SBC), disputada nas ladeiras de paralelepípedo de Ouro Preto. A corrida é uma das mais tradicionais do esporte, completando 100 anos em 2013. Para comemorar o centenário, a FCC pretende integrar dispositivos GPS às charretes, permitindo aos espectadores desfrutarem de dados de telemetria em tempo real.

No mesmo viés de inovação tecnológica, a FCC transmitirá a SBC via satélite para todo o planeta, e quer integrar a telemetria na transmissão, indicando qual seria o vencedor da corrida se as charretes mantivessem suas velocidades até o final da corrida; ela pediu que você escrevesse um programa que, dados as distâncias até a linha de chegada, as velocidades e os números das duas charretes que lideram a corrida, determina quem seria o vencedor da corrida (você pode supor que as charretes não cruzam a linha de chegada simultaneamente).

### Entrada

A entrada consiste de duas linhas; cada linha descreve uma das charretes que lidera a corrida. A descrição de uma charrete consiste de três inteiros  $N$ ,  $D$  e  $V$  indicando, respectivamente, o número da charrete, a sua distância à linha de chegada em metros, e a sua velocidade, em quilômetros por hora. Os números das duas charretes são distintos.

### Saída

Imprima uma única linha, contendo um único número inteiro, indicando o número da charrete que seria vencedora, conforme descrito acima.

### Restrições

- $1 \leq N \leq 99$
- $0 < D \leq 1000$
- $0 < V \leq 50$

### Exemplos:

#### Entrada

45 900 40

17 300 20

#### Saída

17

**Entrada**

1 1000 100

2 1000 99

**Saída**

1



## Pedágio

A invenção do carro tornou muito mais rápido e mais barato realizar viagens de longa distância. Realizar uma viagem rodoviária tem dois tipos de custos: cada quilômetro percorrido na rodovia tem um custo associado (não só devido ao consumo de combustível mas também devido ao desgaste das peças do carro, pneus, etc.), mas também é necessário passar por vários pedágios localizados ao longo da rodovia.

Os pedágios são igualmente espaçados ao longo da rodovia; o começo da estrada não possui um pedágio, mas o seu final pode estar logo após um pedágio (por exemplo, se a distância entre dois pedágios consecutivos for de 37 km e a estrada tiver 111 km, o motorista deve pagar um pedágio aos 37 km, aos 74 km e aos 111 km, logo antes de terminar a sua viagem).

### Tarefa

Dadas as características da rodovia e os custos com gasolina e com pedágios, calcule o custo total da viagem.

### Entrada

A entrada consiste de duas linhas. A primeira linha da entrada contém dois inteiros  $L$  e  $D$  ( $1 \leq L, D \leq 10^4$ ), indicando o comprimento da estrada e a distância entre pedágios, respectivamente. A segunda linha contém dois inteiros  $K$  e  $P$  ( $1 \leq K, P \leq 10^4$ ), indicando o custo por quilômetro percorrido e o valor de cada pedágio. O primeiro pedágio está localizado no quilômetro  $D$  da estrada (ou seja, a distância do início da estrada para o primeiro pedágio é  $D$  quilômetros).

### Saída

Seu programa deve imprimir uma única linha contendo um único inteiro, indicando o custo total da viagem.

Informações sobre a pontuação

- o Em um conjunto de casos de teste que totaliza 40 pontos,  $L, D, K, P \leq 100$ .

### Exemplos:

**Entrada**

**Saída**

*Nome do arquivo: pedagio.x, onde x deve ser c, cpp, java, js ou py*

111 37

141

1 10

---

**Entrada**

**Saída**

100 30

342

3 14

---

**Entrada**

**Saída**

20 70

180

9 17

## Quadrado Mágico

Chama-se de *quadrado mágico* um arranjo, na forma de um quadrado, de  $N \times N$  números inteiros tal que todas as linhas, colunas e diagonais têm a mesma soma.

Por exemplo, o quadrado abaixo:

2	7	6
9	5	1
4	3	8

é um quadrado mágico de soma 15, pois todas as linhas ( $2+7+6 = 15$ ,  $9+5+1 = 15$  e  $4+3+8 = 15$ ), colunas ( $2+9+4 = 15$ ,  $7+5+3 = 15$  e  $6+1+8 = 15$ ) e diagonais ( $2+5+8 = 15$  e  $6+5+4 = 15$ ) têm a mesma soma (15).

### Tarefa

Escreva um programa que, dado um quadrado, determine se ele é mágico ou não e qual a soma dele (caso seja mágico).

### Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha da entrada de cada caso de teste contém um inteiro  $N$  ( $2 < N < 10$ ). As  $N$  linhas seguintes contém  $N$  inteiros cada, separados por exatamente um espaço em branco. Os inteiros dentro do quadrado são todos maiores que 0 (zero) e menores que 1.000.

### Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha com um inteiro representando a soma do quadrado mágico ou -1 caso o quadrado não seja mágico.

### Exemplos:

Entrada	Saída
3	15
2 7 6	
9 5 1	
4 3 8	

**Entrada**

**Saída**

3

-1

1 2 3

4 5 6

7 8 9

---

**Entrada**

**Saída**

4

34

16 3 2 13

5 10 11 8

9 6 7 12

4 15 14 1

## Mapa

Harry ganhou um mapa mágico no qual ele pode visualizar o trajeto realizado por seus amigos. Ele agora precisa de sua colaboração para, com a ajuda do mapa, determinar onde Hermione se encontra.

O mapa tem  $L$  linhas e  $C$  colunas de caracteres, que podem ser '.' (ponto), a letra 'o' (minúscula) ou a letra 'H' (maiúscula). A posição inicial de Hermione no mapa é indicada pela letra 'o', que aparece exatamente uma vez no mapa. A letra 'H' indica uma posição em que Hermione pode ter passado – o mapa é impreciso, e nem toda letra 'H' no mapa representa realmente uma posição pela qual Hermione passou. Mas todas as posições pelas quais Hermione passou são representadas pela letra 'H' no mapa.

A partir da posição inicial de Hermione, Harry sabe determinar a posição atual de sua amiga, apesar da imprecisão do mapa, porque eles combinaram que Hermione somente se moveria de forma que seu movimento apareceria no mapa como estritamente horizontal ou estritamente vertical (nunca diagonal). Além disso, Hermione combinou que não se moveria de forma a deixar que Harry tivesse dúvidas sobre seu caminho (por exemplo, Hermione não passa duas vezes pela mesma posição). Considere o mapa abaixo, com 6 linhas e 7 colunas:

	1	2	3	4	5	6	7
1	.	.	.	H	H	H	.
2	H	H	H	.	.	.	H
3	H	.	H	H	H	.	.
4	H	.	.	.	H	H	.
5	H	.	o	.	.	.	.
6	H	H	H	.	.	H	H

Figura 1

A posição inicial de Hermione no mapa é (5,3)(5,3), e sua posição atual é (4,6)(4,6). As posições marcadas em negrito ('H') são erros no mapa.

Dado um mapa e a posição inicial de Hermione, você deve escrever um programa para determinar a posição atual de Hermione.

### Entrada

A primeira linha contém dois números inteiros  $L$  e  $C$ , indicando respectivamente o número de linhas e o número de colunas. Cada uma das seguintes  $L$  linhas contém  $C$  caracteres.

### Saída

Seu programa deve produzir uma única linha na saída, contendo dois números inteiros: o número da linha e o número da coluna da posição atual de Hermione.

### Restrições

- $2 \leq L \leq 100$
- $2 \leq C \leq 100$
- Apenas os caracteres '.', 'o' e 'H' aparecem no mapa.
- A letra 'o' aparece exatamente uma vez no mapa.
- A letra 'H' aparece ao menos uma vez no mapa.
- O caminho de Hermione está totalmente contido no mapa.
- Na posição da letra 'o' no mapa, há apenas uma letra 'H' como vizinho imediato na vertical ou horizontal.
- Na posição atual de Hermione no mapa, há apenas uma letra 'H' como vizinho imediato na vertical ou horizontal.
- Em cada uma das posições intermediárias do caminho de Hermione, há exatamente duas letras 'H' como vizinhas imediatas na vertical ou horizontal.

### Informações de Pontuação

Em um conjunto de casos de teste somando 20 pontos,  $N \leq 8$

**Exemplos:**

**Entrada**

**Saída**

3 4  
HHHH H... o.HH

1 4

**Entrada**

**Saída**

6 7  
...HHH.  
HHH....  
H.HHH..  
H...HH.  
H.o....  
HHH.HH  
.

4 6

## Robô

Um novo robô de limpeza para um grande salão retangular está sendo desenvolvido. O robô vai percorrer o caminho definido por uma linha marcada no chão, que é coberto com ladrilhos quadrados, brancos e pretos: ladrilhos pretos indicam o caminho que o robô deve percorrer. Ao movimentar-se, o robô pode andar apenas em linha reta, para a frente. Parado, o robô pode girar para as quatro direções (Norte, Sul, Leste e Oeste).

Dados um mapa indicando a cor de cada ladrilho no chão e a posição inicial do robô, você deve escrever um programa que determine a posição final do robô.

### Entrada

A primeira linha contém dois inteiros  $L$  e  $C$  indicando as dimensões do salão (número de linhas e número de colunas), medidas em ladrilhos. A segunda linha contém dois inteiros  $A$  e  $B$  indicando respectivamente a linha e a coluna da posição inicial do robô (as linhas são numeradas de 1 a  $L$ , de cima para baixo; as colunas são numeradas de 1 a  $C$ , da esquerda para a direita). Cada uma das  $L$  linhas seguintes contém  $C$  inteiros, zeros ou uns. Nessa representação, o valor "1" indica que o ladrilho correspondente é preto. O ladrilho da linha  $A$  e coluna  $B$  sempre é preto. O caminho do robô é definido unicamente: em nenhum momento o robô necessita fazer uma escolha sobre em qual direção ir (em outras palavras, todo ladrilho preto tem no máximo dois vizinhos pretos e o ladrilho inicial tem um vizinho preto).

### Saída

Seu programa deve imprimir apenas uma linha, contendo dois números inteiros, respectivamente a linha e a coluna da posição final do robô.

### Restrições

- o  $1 \leq L, C \leq 1000$
- o  $1 \leq A \leq L$
- o  $1 \leq B \leq C$
- o A posição final é diferente da posição inicial.

### Exemplos:

#### Entrada

3 5

1 1

1 0 0 0 1

#### Saída

1 5



1 0 0 1 1

1 1 1 1 0

---

**Entrada**

**Saída**

4 7

4 2

3 4

0 0 1 1 1 1 1

1 1 1 0 0 0 1

1 0 0 1 0 1 1

1 1 0 1 1 1 0

## Soma

Temos uma sequência de  $N$  quadrados desenhados lado a lado. Cada quadrado possui um número natural anotado dentro dele. Dados a sequência dos  $N$  quadrados e um valor  $K$ , quantos retângulos distintos existem cuja soma dos números dentro do retângulo é exatamente igual a  $K$ ? Por exemplo, a figura mostra uma sequência de  $N=10$  quadrados para a qual existem 5 retângulos cuja soma dos números é igual a  $K=4$ .

2	0	1	1	0	0	8	4	1	3
2	0	1	1	0	0	8	4	1	3
2	0	1	1	0	0	8	4	1	3
2	0	1	1	0	0	8	4	1	3
2	0	1	1	0	0	8	4	1	3

### Entrada

A primeira linha da entrada contém dois inteiros  $N$  e  $K$  representando o número de quadrados na sequência e o valor da soma desejada. A segunda linha da entrada contém  $N$  números naturais  $X_i$ , para  $1 \leq i \leq N$ , indicando a sequência de números anotados dentro dos quadrados.

### Saída

Seu programa deve imprimir uma linha contendo um número inteiro representando quantos retângulos existem na sequência cuja soma é igual a  $K$ .

### Restrições

- o  $1 \leq N \leq 500000$  ( $5 \times 10^5$ )
- o  $0 \leq K \leq 10^6$
- o  $0 \leq X_i \leq 100$  para  $1 \leq i \leq N$

**Informações sobre a pontuação**

- o Em um conjunto de casos de teste somando 10 pontos,  $N \leq 50$
- o Em um conjunto de casos de teste somando 20 pontos,  $N \leq 10^4$
- o Em um conjunto de casos de teste somando 30 pontos,  $K > 0$  e  $X_i > 0$  para  $1 \leq i \leq N$
- o Em um conjunto de casos de teste somando 40 pontos, nenhuma restrição adicional (note que para esta subtarefa o inteiro da saída pode não caber em 32 bits).

**Exemplos:****Entrada**

10 4

2 0 1 1 0 0 8 4 1 3

**Saída**

5

**Entrada**

15 0

0 0 0 0 0 5 12 0 1 0 0 0 51 0 0

**Saída**

25

## Copa do Mundo

A Nlogônia é atualmente um dos países com maior crescimento econômico no mundo, e seus governantes têm se esforçado para que o país seja mais conhecido e respeitado internacionalmente. Recentemente a Nlogônia foi escolhida para ser a sede da Copa do Mundo de Futebol Amador, e está se preparando para receber os milhares de torcedores que o evento atrai.

Como parte da preparação para a Copa, o governo planeja realizar uma reforma em todo o sistema de transporte intermunicipal, que é hoje composto de uma malha de rodovias e ferrovias, cada rodovia ou ferrovia interligando um par de cidades. Com as rodovias e ferrovias existentes já é possível viajar entre qualquer par de cidades (possivelmente passando por outras cidades no caminho), mas o governo quer oferecer melhores condições de transporte para os visitantes e a população.

Como não há recursos para reformar todas as rodovias e ferrovias, o governo quer escolher um conjunto de rodovias e ferrovias para ser reformado, e já realizou um estudo para estabelecer o custo de reforma de cada rodovia e ferrovia. A escolha deve obedecer aos seguintes critérios:

1. Ao final da reforma, deve ser possível viajar entre qualquer par de cidades (possivelmente passando por outras cidades) utilizando apenas rodovias ou ferrovias reformadas;
2. Para priorizar o transporte público, dentre as escolhas que satisfazem a restrição 1, deve-se escolher uma que minimize o número de rodovias reformadas;
3. Dentre as escolhas que satisfazem as restrições 1 e 2, deve-se escolher uma que minimize o custo total.

Você foi contratado para escrever um programa que, conhecidos os custos de reforma de cada rodovia e ferrovia, determine o menor custo possível para a reforma, obedecendo os critérios estabelecidos.

### Entrada

A primeira linha da entrada contém três inteiros  $N$ ,  $F$  e  $R$ , indicando respectivamente o número de cidades, de ferrovias e de rodovias. As cidades são identificadas por inteiros de 1 a  $N$ . Cada uma das  $F$  linhas seguintes descreve uma ferrovia e contém três inteiros  $A$ ,  $B$  e  $C$ , onde  $A$  e  $B$  representam cidades e  $C$  representa o custo da reforma da ferrovia que interliga  $A$  e  $B$ . Cada uma das  $R$  linhas seguintes descreve uma rodovia e contém três inteiros  $I$ ,  $J$  e  $K$ , onde  $I$  e  $J$  representam cidades e  $K$  representa o custo da reforma da rodovia que interliga  $I$  e  $J$ .

### Saída

Seu programa deve produzir uma única linha, contendo o menor custo possível para o conjunto de reformas de ferrovias e rodovias, obedecendo aos critérios estabelecidos.

**Restrições**

- o  $2 \leq N \leq 100$ ;  $1 \leq F \leq N(N-1)/2$ ;  $1 \leq R \leq N(N-1)/2$
- o  $1 \leq A < B \leq N$  e  $1 \leq I < J \leq N$
- o  $1 \leq C \leq 1000$  e  $1 \leq K \leq 1000$

**Informações sobre a pontuação**

- o Para um conjunto de casos de testes totalizando 20 pontos,  $2 \leq N \leq 6$ .

**Exemplos:**

Entrada	Saída
3 3 2	1900

1 2 1000

1 3 1000

2 3 900

1 3 800

2 3 700

---

Entrada	Saída
5 4 5	1050

3 4 300

1 2 100

2 4 300

1 3 250

4 5 600

3 4 200

2 3 100

2 5 400

1 5 450

---

Entrada	Saída
5 2 3	220
4 5 60	
2 3 60	
1 2 50	
1 4 50	
3 4 50	

## Gincana

As duas turmas do terceiro ano de sua escola realizam anualmente uma gincana. Nessa gincana, a delegação de cada turma é dividida em grupos de  $K$  pessoas, de forma que  $K$  seja o maior número possível que divida as duas delegações sem que sobre alguém. Depois, os grupos competem uns com os outros, ganhando pontos para determinar a turma vencedora. Sua turma pode levar qualquer número  $X$  de pessoas entre 1 e  $M$ , a quantidade de alunos na turma, e você sabe que a turma rival levará exatamente  $N$  pessoas para a gincana. Os integrantes da sua turma são muito bons em competições individuais, mas não trabalham bem em equipe. Portanto, é sua tarefa encontrar a maior delegação possível que sua turma pode levar à competição para que a gincana aconteça com grupos de  $K = 1$  pessoa. Por exemplo, se  $N = 9$  e  $M = 6$  a sua turma deve levar uma delegação de  $X = 5$  pessoas, já que, para esse valor, a única divisão possível é em grupos de  $K = 1$  pessoa e, para  $X = 6$ , os grupos seriam de 3 pessoas.

### Entrada

A primeira e única linha contém dois inteiros  $N$  e  $M$ , representando respectivamente o tamanho da delegação rival e o tamanho da sua turma.

### Saída

Seu programa deve produzir uma única linha, contendo um inteiro  $X$ , o maior tamanho possível da delegação da sua turma para o qual a gincana aconteça com grupos de uma pessoa.

### Restrições

- o  $1 \leq N, M \leq 10^{18}$ .

### Informações sobre a pontuação

- o Em um conjunto de casos de teste equivalente a 40 pontos,  $N, M \leq 10^3$ .
- o Em um conjunto de casos de teste equivalente a 60 pontos,  $N, M \leq 10^5$ .
- o Em um conjunto de casos de teste equivalente a 80 pontos,  $N, M \leq 10^7$ .

### Exemplos:

Entrada	Saída
9 6	5

Entrada	Saída
6 9	7

Entrada	Saída
6 3	1

Entrada	Saída
2310 126	113