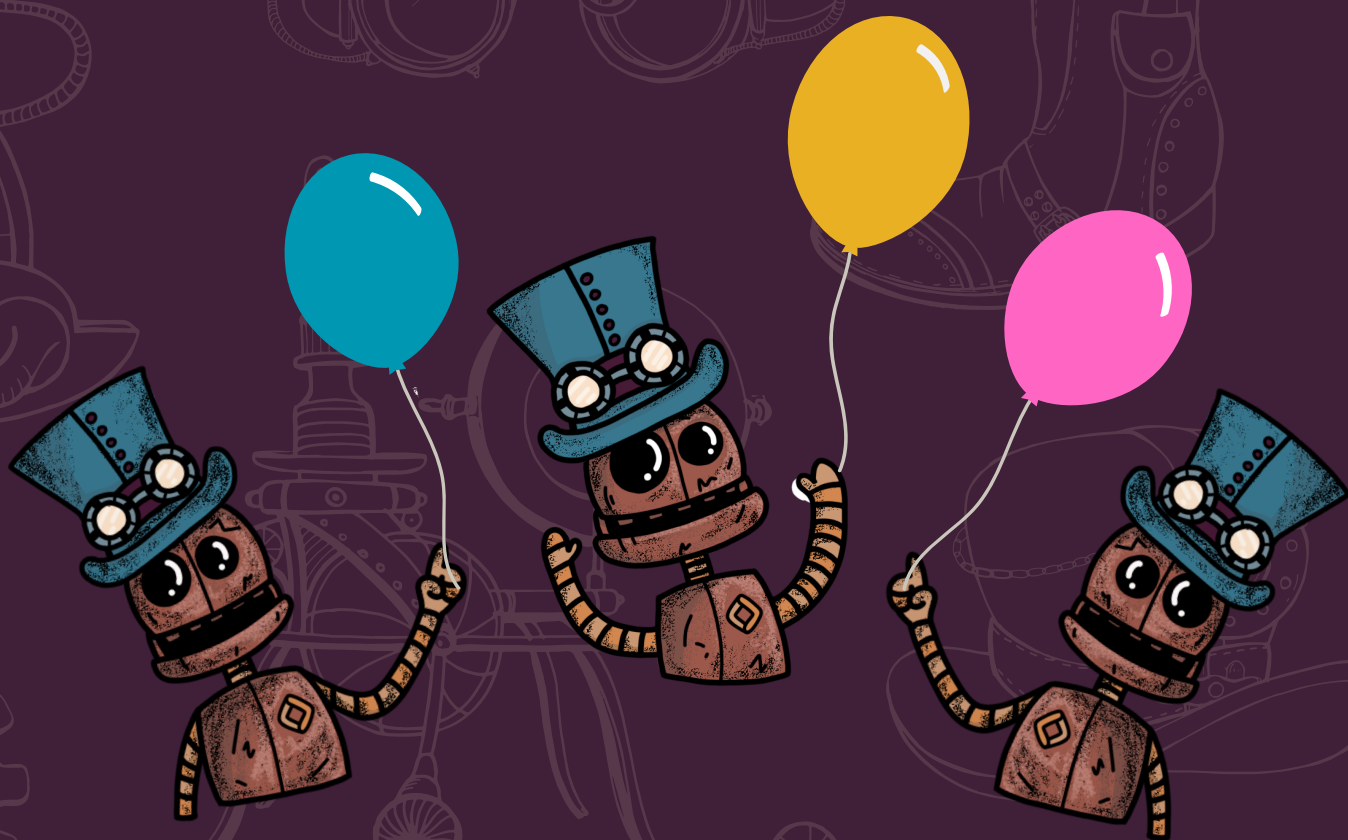


# HACKTOBERTFEST OPENDEV UFCG COMPETIÇÃO GERAL DE PROGRAMAÇÃO - MISTA



28 OUT 2023

UFCG - CAMPUS CAMPINA GRANDE



# Problem A. Quantidade de Ímpares

**Time limit** 2000 ms

**Mem limit** 1048576 kB

## Problema

Neste problema, um arquivo de entrada contém vários casos de teste.

Primeiro você recebe um número inteiro  $T$ . Resolva o seguinte problema para os  $T$  casos de teste.

- Temos  $N$  inteiros positivos  $A_1, A_2, \dots, A_N$ . Quantos deles são ímpares?

## Restrições

- $1 \leq T \leq 100$
- $1 \leq N \leq 100$
- $1 \leq A_i \leq 10^9$
- Todos os valores na entrada são números inteiros.

## Entrada

A entrada é fornecida pela entrada padrão no seguinte formato, onde  $\text{test}_i$  representa o  $i$ -ésimo caso de teste:

```
T
test1
test2
⋮
testT
```

Cada caso de teste está no seguinte formato:

```
N
A1 A2 ... AN
```

## Saída

Imprima  $T$  linhas. A  $i$ -ésima linha deve conter a resposta para o  $i$ -ésimo caso de teste.

## Caso de Teste 1

Entrada	Saída
4 3 1 2 3 2 20 23 10 6 10 4 1 5 9 8 6 5 1 1 1000000000	2 1 5 0

Esta entrada contém quatro casos de teste.

A segunda e terceira linhas correspondem ao primeiro caso de teste, onde  $N = 3$ ,  $A_1 = 1$ ,  $A_2 = 2$ ,  $A_3 = 3$ .

Temos dois números ímpares em  $A_1$ ,  $A_2$  e  $A_3$ , então a primeira linha deve conter 2.

# Problem B. Maratona Universal

**Time limit** 2000 ms

**Mem limit** 1048576 kB

## Problema

Está acontecendo a maratona universal de programação. Como esse é um evento de proporções inimagináveis,  $10^{16}$  problemas foram preparados para desafiar programadores de todo o universo.

Os IDs dos problemas são definidos da seguinte forma, na ordem do primeiro ao último: **A** , **B** , ..., **Z** , **AA** , **AB** , ..., **ZZ** , **AAA** , ...

Em outras palavras, os IDs são definidos na seguinte ordem:

- strings de tamanho 1 consistindo de letras maiúsculas, em ordem lexicográfica;
- strings de tamanho 2 consistindo de letras maiúsculas, em ordem lexicográfica;
- strings de tamanho 3 consistindo de letras maiúsculas, em ordem lexicográfica;
- ...

Dada uma string  $S$  representando o ID de um problema da maratona, encontre o índice do problema. (Veja os exemplos para melhor entendimento)

## Restrições

- $S$  é um ID de problema válido para a maratona.

## Entrada

A entrada é dada no seguinte formato:

$S$

## Saída

Imprima o índice inteiro correspondente do problema.

### Caso de teste 1

Input	Output
AB	28

O problema cujo ID é **AB** é o 28<sup>a</sup> problema da maratona, então a resposta deve ser 28.

### Caso de teste 2

Input	Output
C	3

O problema cujo ID é **C** é o 3ª problema da maratona, então a resposta deve ser 3.

### Caso de teste 3

Input	Output
BRUTMHYHIIZP	100000000000000000

O problema cujo ID é **BRUTMHYHIIZP** é o último problema da maratona, então a resposta deve ser  $10^{16}$ .

# Problem C. Artista

**Time limit** 2000 ms

**Mem limit** 1048576 kB

## Problema

Você recebe uma matriz  $H \times W$   $A$  que consiste em números inteiros entre 0 e 26. O elemento na  $i$ -ésima linha do topo e na  $j$ -ésima coluna da esquerda é  $A_{i,j}$ .

Sejam  $S_1, S_2, \dots, S_H$   $H$  strings de comprimento  $W$  que satisfazem o seguinte:

- O  $j$ -ésimo caractere de  $S_i$  é um ponto final ( . ) se  $A_{i,j}$  é 0, e a  $A_{i,j}$ -ésima letra do alfabeto caso contrário. (Por exemplo, a 4-ésima letra é **D** .)

Imprima  $S_1, S_2, \dots, S_H$  em ordem.

## Restrições

- $1 \leq H \leq 100$
- $1 \leq W \leq 100$
- $0 \leq A_{i,j} \leq 26$
- Todos os valores na entrada são números inteiros.

## Entrada

A entrada é fornecida pela entrada padrão no seguinte formato:

```
H W
A1,1 A1,2 ... A1,W
A2,1 A2,2 ... A2,W
⋮
AH,1 AH,2 ... AH,W
```

## Saída

Imprima  $H$  linhas. A  $i$ -ésima linha deve conter  $S_i$ .

## Caso de Teste 1

Entrada	Saída
<pre>2 3 0 1 2 0 0 3</pre>	<pre>.AB ..C</pre>

Temos  $S_1 = \text{.AB}$  e  $S_2 = \text{..C}$ . Imprima-os em ordem.

**Caso de Teste 2**

Entrada	Saída
3 3 24 0 0 0 25 0 0 0 26	X.. .Y.. ..Z

**Caso de Teste 3**

Entrada	Saída
3 1 2 9 4	B I D

# Problem D. Contagem de Caminhos

**Time limit** 2000 ms

**Mem limit** 1048576 kB

## Problema

Você é o presidente da LinearLândia e está analisando o mapa do seu país, ele contém  $N$  cidades numeradas de 1 a  $N$  e  $M$  estradas numeradas de 1 a  $M$ . A estrada  $i$  conecta as cidades  $u_i$  e  $v_i$ . Cada cidade se conecta diretamente com, no máximo, 10 outras cidades. Uma estrada pode ser percorrida em ambos os sentidos, ou seja, se existe uma estrada conectando as cidades  $u_i$  e  $v_i$ , é possível fazer uma rota iniciando em  $u_i$  e finalizando em  $v_i$  ou vice-versa.

Digamos que  $K$  seja o número de caminhos simples (caminhos sem cidades repetidas) Iniciando na cidade 1. Imprima  $\min(K, 10^6)$ .

## Restrições

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq M \leq \min\left(2 \times 10^5, \frac{N(N-1)}{2}\right)$
- $1 \leq u_i, v_i \leq N$
- O mapa fornecido representa um grafo simples.
- Cada cidade se conecta com, no máximo, 10 outras cidades.
- Todos os valores na entrada são inteiros.

## Entrada

A entrada é fornecida no seguinte formato:

```
N M
u1 v1
u2 v2
⋮
uM vM
```

## Saída

Imprima a resposta.

## Caso de teste 1

Input	Output
4 2 1 2 2 3	3



Nós temos 3 caminhos simples. (Note que um caminho de tamanho 0 também é contabilizado.)

- Cidade 1;
- Cidade 1, cidade 2;
- Cidade 1, cidade 2, cidade 3.

**Caso de teste 2**

Input	Output
4 6 1 2 1 3 1 4 2 3 2 4 3 4	16

**Caso de teste 3**

Input	Output
8 21 2 6 1 3 5 6 3 8 3 6 4 7 4 6 3 4 1 5 2 4 1 2 2 7 1 4 3 5 2 5 2 3 4 5 3 7 6 7 5 7 2 8	2023

# Problem E. Sequência de Strings

**Time limit** 2000 ms

**Mem limit** 1048576 kB

## Problema

Você recebe  $N$  strings  $S_1, S_2, \dots, S_N$  nesta ordem.

Imprima  $S_N, S_{N-1}, \dots, S_1$  nesta ordem.

## Restrições

- $1 \leq N \leq 10$
- $N$  é um número inteiro.
- $S_i$  é uma string de comprimento entre 1 e 10, inclusive, consistindo de letras minúsculas, letras maiúsculas e dígitos.

## Entrada

A entrada é fornecida pela entrada padrão no seguinte formato:

```
N
S1
S2
⋮
SN
```

## Saída

Imprima  $N$  linhas. A  $i$ -ésima ( $1 \leq i \leq N$ ) linha deve conter  $S_{N+1-i}$ .

## Caso de Teste 1

Entrada	Saída
3 Takahashi Aoki Snuke	Snuke Aoki Takahashi

Nós temos  $N = 3$ ,  $S_1 = \text{Takahashi}$ ,  $S_2 = \text{Aoki}$ , and  $S_3 = \text{Snuke}$ .

Assim, você deve imprimir **Snuke**, **Aoki** e **Takahashi** nesta ordem.

## Caso de Teste 2

Entrada	Saída
4 2023 Year New Happy	Happy New Year 2023

As strings fornecidas podem conter dígitos.

# Problem F. Operações de Subtração

**Time limit** 2000 ms

**Mem limit** 1048576 kB

## Problema

Você tem dois inteiros positivos  $A$  e  $B$ .

Você irá repetir a seguinte operação até que  $A = B$ :

- comparar  $A$  e  $B$  e realizar uma das duas operações:
  - Se  $A > B$ , substituir  $A$  por  $A - B$ ;
  - Se  $A < B$ , substituir  $B$  por  $B - A$ .

Quantas operações serão realizadas para que obtenhamos  $A = B$ ? É garantido que uma quantidade finita de operações irá resultar em  $A = B$ .

## Restrições

- $1 \leq A, B \leq 10^{18}$
- Todos os valores na entrada são inteiros.

## Entrada

A entrada é fornecida no seguinte formato:

$A$   $B$

## Saída

Imprima a resposta.

### Caso de teste 1

Input	Output
3 8	4

Inicialmente,  $A = 3$  e  $B = 8$ . As operações ocorrem da seguinte forma:

- $A < B$ , então substitui  $B$  por  $B - A = 5$ , obtendo  $A = 3$  e  $B = 5$ .
- $A < B$ , então substitui  $B$  por  $B - A = 2$ , obtendo  $A = 3$  e  $B = 2$ .
- $A > B$ , então substitui  $A$  por  $A - B = 1$ , obtendo  $A = 1$  e  $B = 2$ .
- $A < B$ , então substitui  $B$  por  $B - A = 1$ , obtendo  $A = 1$  e  $B = 1$ .

Portanto, foram necessárias 4 operações.

**Caso de teste 2**

Input	Output
1234567890 1234567890	0

Note que a entrada pode exceder o limite de 32 bits.

**Caso de teste 3**

Input	Output
1597 987	15

# Problem G. Máquina de Algodão Doce

**Time limit** 3000 ms

**Mem limit** 1048576 kB

## Problema

Existem  $N$  pessoas enumeradas de 1 a  $N$  que irão visitar um hotel. Para cada pessoa, sabemos o horário de entrada e quanto tempo a mesma vai passar no hotel. A pessoa  $i$  entra no hotel  $T_i$  minutos depois da abertura do hotel e sai depois de  $D_i$  minutos.

Durante sua estadia no hotel, cada pessoa tem o direito de pedir um (apenas um, durante toda a estadia) algodão doce na recepção. Porém, quando alguém pede um algodão doce, a máquina demora exatamente um minuto para produzir outro algodão doce. Particularmente, duas pessoas não podem pedir um algodão doce no mesmo minuto. Para simplificar, vamos considerar que cada pessoa só pode pedir um algodão doce em tempos inteiros (em minutos) e que uma pessoa pode pedir um algodão doce no último instante de sua estadia. Por exemplo, se uma pessoa entra no minuto 3 e sai 2 minutos depois, ela pode pedir o algodão doce nos minutos 3, 4 e 5.

Qual é a maior quantidade de **pessoas** que vão conseguir comer um algodão doce durante sua estadia do hotel considerando que elas vão agir de maneira ótima?

## Restrições

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq T_i, D_i \leq 10^{18}$
- Todos os valores na entrada são inteiros.

## Entrada

A entrada será dada no seguinte formato

```
N
T1 D1
T2 D2
⋮
TN DN
```

## Saída

Imprima na saída a maior quantidade de pessoas que vai conseguir pedir um algodão doce.

## Caso de teste 1

Input	Output
5 1 1 1 1 2 1 1 2 1 4	4

Por exemplo, conseguimos fazer 4 pessoas pedirem um algodão da seguinte forma:

- Minuto 1 : Pessoas 1, 2, 4, 5 estão no hotel. A pessoa 4 pega um algodão doce .
- Minuto 2 : Pessoa 3 entra no hotel. A pessoa 1 pega um algodão doce. As pessoas 1, 2 saem do hotel.
- Minuto 3 : A pessoa 3 pega um algodão doce. As pessoas 3, 4 saem do hotel.
- Minuto 4 : A pessoa 5 pega um algodão doce.
- Minuto 5 : A pessoa 5 sai do hotel.

É impossível todos pegarem um algodão doce. Portanto, a melhor resposta nesse caso é 4.

#### Caso de teste 2

Input	Output
2 1 1 10000000000000000000 10000000000000000000	2

#### Caso de teste 3

Input	Output
10 4 1 1 2 1 4 3 2 5 1 5 1 4 1 2 1 4 1 2 4	6

# Problem H. Colisões

**Time limit** 2000 ms

**Mem limit** 1048576 kB

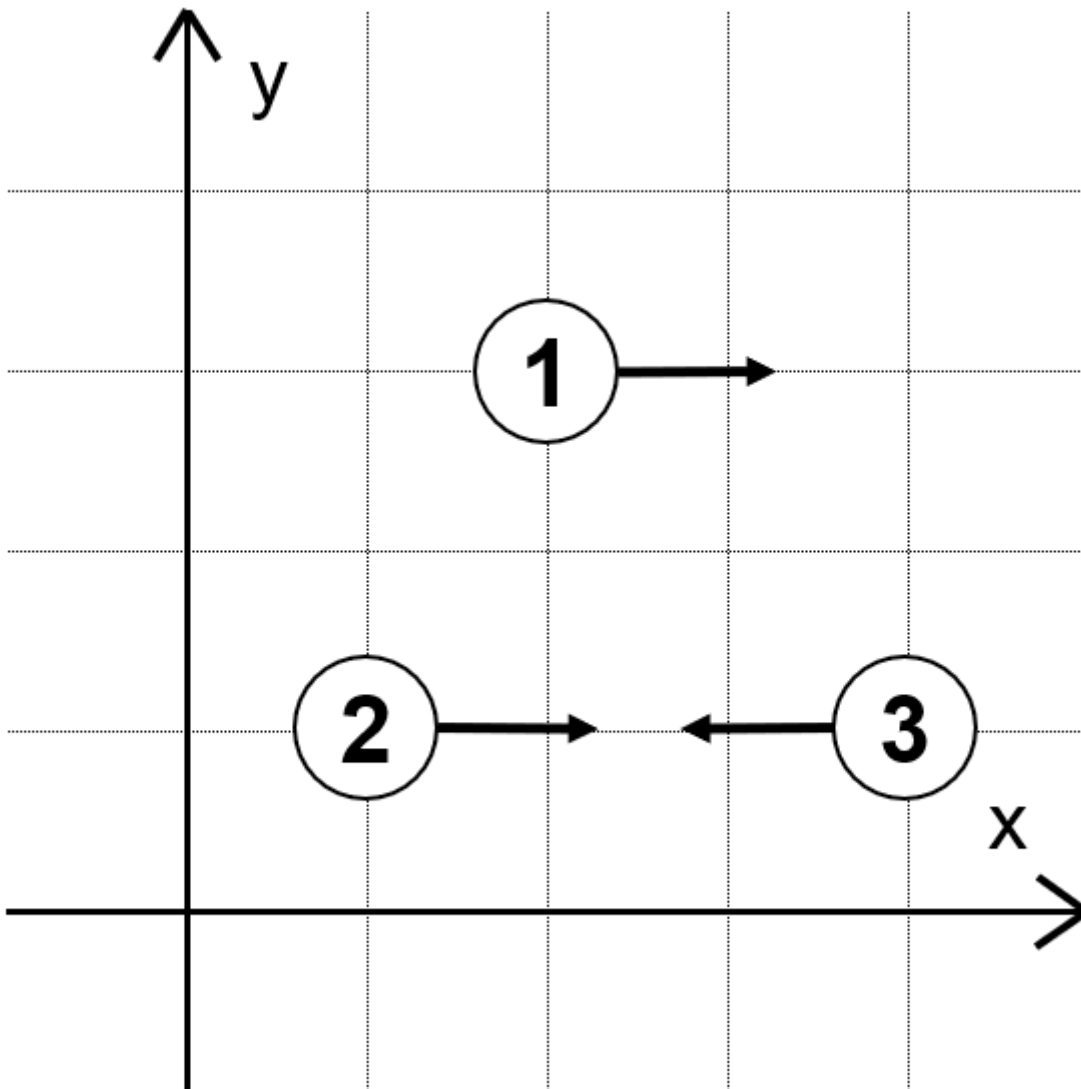
## Problema

Existem  $N$  pessoas em um plano cartesiano. A pessoa  $i$  está na coordenada  $(X_i, Y_i)$ . As posições das pessoas são completamente diferentes, ou seja, é impossível duas pessoas estarem na mesma localização.

Nós temos uma string  $S$  de tamanho  $N$  consistindo de **L** e **R**.

Se  $S_i = \text{R}$ , a pessoa  $i$  está olhando para a direita; se  $S_i = \text{L}$ , a pessoa  $i$  está olhando para a esquerda. Todas as pessoas começam a andar simultaneamente para a direção que estão olhando.

Por exemplo, a figura abaixo mostra o movimento das pessoas quando  $(X_1, Y_1) = (2, 3), (X_2, Y_2) = (1, 1), (X_3, Y_3) = (4, 1), S = \text{RRL}$ .





Nós dizemos que existe uma colisão quando duas pessoas andando em direções opostas passam a ocupar a mesma posição no plano cartesiano. Supondo que todas as pessoas andem por um tempo infinito, irá existir alguma colisão?

### Restrições

- $2 \leq N \leq 2 \times 10^5$
- $0 \leq X_i \leq 10^9$
- $0 \leq Y_i \leq 10^9$
- $(X_i, Y_i) \neq (X_j, Y_j)$  if  $i \neq j$ .
- Todos  $X_i$  e  $Y_i$  são inteiros.
- $S$  é uma string de tamanho  $N$  consistindo de **L** e **R**.

### Entrada

A entrada é fornecida no seguinte formato:

```
N
X1 Y1
X2 Y2
⋮
XN YN
S
```

### Saída

Se irá ocorrer alguma colisão, imprima **Yes** ; caso contrário, imprima **No** .

#### Caso de teste 1

Input	Output
3 2 3 1 1 4 1 RRL	Yes

Esse caso de teste corresponde a imagem do enunciado do problema

Se todas as pessoas continuarem andando, as pessoas 2 e 3 irão colidir. Portanto, a resposta deve ser **Yes** .

#### Caso de teste 2

Input	Output
2 1 1 2 1 RR	No

Como as pessoas 1 e 2 andam na mesma direção, elas nunca irão colidir.

**Caso de teste 3**

Input	Output
10 1 3 1 4 0 0 0 2 0 4 3 1 2 4 4 2 4 4 3 3 RLRRRLRLRR	Yes