

XOR

O João é um aluno que está realizando ao juntar duas de suas paixões: os computadores e a lógica. Ele iniciou recentemente os estudos de programação e o assunto não sai de sua cabeça. Recentemente, o João estava revendo mentalmente os conceitos fundamentais da programação e lembrando de conceitos mais avançados de lógica, ele notou um detalhe. . .

O “ou exclusivo” (*exclusive OR - XOR*) é uma operação lógica muito semelhante ao OU tradicional, mas possui uma diferença: é falso quando os dois operandos são verdadeiros. A seguir está a tabela verdade para o XOR:

A	B	A XOR B
V	V	F
V	F	V
F	V	V
F	F	F

João estava aprendendo a linguagem de programação C, ele lembrava que o E (AND) possuía o operador tradicional `&&` e a sua versão bit-a-bit (bitwise) `&`. Mas não conseguia lembrar o operador lógico do XOR. Ele nem sabia se o operador existia ou não. . . Ele sabia que uma versão bit-a-bit do XOR não funcionaria para ele! Neste exercício, você precisa combinar os operadores, instruções condicionais e de repetição que precisar para produzir um programa que produza o resultado da operação XOR para quantas vezes o usuário desejar (mas, com certeza, menos de 50). . .

Entrada

A entrada deve ser lida da entrada padrão e possui o seguinte formato: a primeira linha da entrada contém um único inteiro N , indicando o número de operações a serem calculadas. As próximas N linhas contém dois números inteiros “ XY ”, separados por espaço, que podem ser 1 (falso) ou 2 (verdadeiro).

Saída

A saída é composta por N linhas, indicando o resultado da operação XOR lida anteriormente: 1 se for falso, 2 se for verdadeiro.

Restrições

- $1 \leq N < 50$
- X e $Y = 1$ ou 2

Exemplo de Entrada 1

```
3
1 1
1 1
1 2
```

Exemplo de Saída 1

```
1
1
2
```

Exemplo de Entrada 2

```
5
2 1
1 2
1 2
1 1
```

1 1

Exemplo de Saída 2

2

2

2

1

1

Author: Daniel Sundfeld <daniel.sundfeld@unb.br>