# **Corrida de Charrete**

(OBI2011) A Federação de Corridas de Charrete (FCC) organiza todo ano a Subida Brigite 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.

### **Exemplos**

| **Entrada** | **Saída** |
| --- | --- |
| 45 900 40 17 300 20 | 17 |
| 1 1000 100 2 1000 99 | 1 |

# **Campeonato de Atletismo**

(OBI2012) Leonardo é um corredor profissional que participa de diversos campeonatos de atletismo pelo mundo. O tamanho das pistas ao redor do mundo não é padronizado. Por isso, Leonardo, que treina em um clube que possui uma pista circular, resolveu fixar seu treinamento em *C* metros, ao invés de um número fixo de voltas na pista. Após cada treinamento, Leonardo deve tomar meio litro de água antes de fazer qualquer esforço, e por isso quer deixar sua garrafa de água exatamente no ponto da pista onde ele termina o seu treinamento.

Sabendo o comprimento da pista de corrida que Leonardo pretende treinar, ele resolveu pedir sua ajuda para calcular o local do *ponto de término* do treinamento. O ponto de término é o local da pista onde ele termina o percurso de *C* metros considerando que ele parte do ponto de partida e se movimenta sempre na mesma direção. O ponto de término é dado pelo número de metros entre o ponto de partida e o local onde Leonardo termina seu treinamento, contados na direção do percurso. Leonardo quer deixar sua garrafa de água neste ponto.

Por exemplo, se a pista tem 12 metros e Leonardo fixou seu treinamento em 22 metros, o ponto de término é 10.

Sua tarefa é, dado o número *C* de metros que Leonardo pretende correr e o comprimento *N* em metros da pista circular, determinar o ponto de término de seu treinamento.

### **Entrada**

A entrada consiste em apenas uma linha contendo dois inteiros *C* e *N* que indicam, respectivamente, o número de metros que Leonardo pretende correr e o comprimento da pista.

### **Saída**

Seu programa deve imprimir apenas uma linha, contendo apenas um inteiro, indicando o ponto de término do treinamento de Leonardo.

### **Restrições**

* 1 ≤ *C* ≤ 108
* 1 ≤ *N* ≤ 100

### **Exemplos**

| **Entrada** | **Saída** |
| --- | --- |
| 7000 100 | 0 |
| 918 76 | 6 |

## **Garçom**

OBI2010 - Junior - Fase 1) Parte do treinamento de um novo garçom é carregar uma grande bandeja com várias latas de bebidas e copos e entregá-las todas numa mesa do restaurante. Durante o treinamento é comum que os garçons deixem cair as bandejas, quebrando todos os copos.

A SBC -- Sociedade Brasileira de Copos -- analisou estatísticas do treinamento de diversos garçons e descobriu que os garçons em treinamento deixam cair apenas bandejas que têm mais latas de bebidas que copos. Por exemplo, se uma bandeja tiver 10 latas e 4 copos, certamente o garçom em treinamento a deixará cair, quebrando os 4 copos. Já se a bandeja tiver 5 latas e 6 copos, ele conseguirá entregá-la sem deixar cair.

Escreva um programa que, dado o número de latas e copos em cada bandeja que o garçom tentou entregar, imprime o total de copos que ele quebrou.

### **Entrada**

A primeira linha da entrada contém um inteiro N representando o número de bandejas que o garçom tentou entregar. As N linhas seguintes representam as N bandejas. Cada linha contém dois inteiros L e C, indicando o número de latas e o número de copos naquela bandeja, respectivamente.

### **Saída**

Seu programa deve imprimir uma única linha, contendo um único inteiro, indicando o número total de copos que o garçom quebrou.

### **Restrições**

* 1 ≤ N ≤ 100
* 0 ≤ L, C ≤ 100

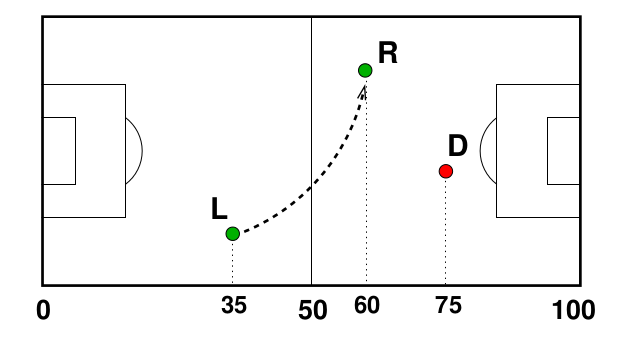
### **Exemplos**

| **Entrada**  3 10 5 6 8 3 3 | **Saída**  5 |
| --- | --- |

| **Entrada**  4 10 6 8 8 5 1 100 100 | **Saída**  7 |
| --- | --- |

# **Impedimento!**

(OBI2015 - Nível 1 - Fase 2) A regra do impedimento no futebol pode parecer estranha, mas sem ela, se a gente pensar bem, o jogo ficaria muito chato! Ela funciona dadas as posições de três jogadores: L o jogador atacante que lança a bola; R o jogador atacante que recebe a bola; e D o último jogador defensor. E a regra vale somente se o jogador R está no seu campo de ataque; se o jogador R está no seu campo de defesa ou na linha divisória do meio campo, ele não está em impedimento. Neste problema o campo tem 100 metros de comprimento. Dadas as posições desses três jogadores, no momento exato do lançamento, haverá impedimento se e somente se a seguinte condição for verdadeira: **(R>50) e (L< R) e (R >D)**.

A regra parece estranha, não é mesmo? Mas a gente nem precisa entender a lógica dela. O seu programa deve apenas determinar, dadas as três posições L,R e D, se há ou não impedimento, implementando exatamente a condição acima. A figura abaixo mostra um exemplo onde \bf não há impedimento:

### **Entrada**

A entrada é composta de apenas uma linha, contendo os três inteiros L, R e D.

### **Saída**

Seu programa deve produzir uma única linha, contendo um único caractere, que deve ser "S" caso haja impedimento, ou "N" caso contrário.

### **Exemplos**

| **Entrada** | **Saída** |
| --- | --- |
| 35 60 75 | N |
| 55 68 67 | S |
| 66 80 80 | N |