

# Problema A: Copa do Mundo

---

Uma Copa do Mundo de futebol de botões está sendo realizada com times de todo o mundo. A classificação é baseada no número de pontos ganhos pelos times, e a distribuição de pontos é feita da forma usual. Ou seja, quando um time ganha um jogo, ele recebe 3 pontos; se o jogo termina empatado, ambos os times recebem 1 ponto; e o perdedor não recebe nenhum ponto.

Dada a classificação atual dos times e número de times participantes na Copa do Mundo, sua tarefa é de determinar quantos jogos terminaram empatados até o momento.

## ENTRADA

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém dois inteiros  $T$  e  $N$ , indicando respectivamente o número de times participantes ( $2 \leq T \leq 200$ ) e o número de partidas jogadas ( $0 \leq N \leq 10000$ ). Cada uma das  $T$  linhas seguintes contém o nome de um time (uma cadeia de no máximo 10 letras e dígitos), seguido de um espaço em branco, seguido do número de pontos que o time obteve até o momento. O final da entrada é indicado por uma linha que contém apenas o número zero. Os dados devem ser lidos da entrada padrão.

## SAÍDA

Para cada caso de teste, seu programa deve imprimir uma única linha contendo um número inteiro, representando a quantidade de jogos que terminaram empatados até o momento. O resultado de seu programa deve ser escrito na saída padrão.

### EXEMPLO DE ENTRADA

```
3 3
Brasil 3
Australia 3
Croacia 3
3 3
Brasil 3
Japao 1
Australia 1
0 0
```

### EXEMPLO DE SAÍDA

```
0
2
```

## Problema B: Suco de Acerola

---

Natural das Antilhas, a acerola já era apreciada pelos nativos das Américas há muitos séculos. Mas o grande interesse por essa fruta surgiu na década de 1940, quando cientistas porto-riquenhos descobriram que a acerola contém grande quantidade de vitamina C. A acerola apresenta, em uma mesma quantidade de polpa, até 100 vezes mais vitamina C do que a laranja e o limão, 20 vezes mais do que a goiaba e até 10 vezes mais do que o caju e a amora.

Um grupo de amigos está visitando o Sítio do Picapau Amarelo, renomado produtor de acerola. Com a permissão de Dona Benta, dona do sítio, colheram uma boa quantidade de frutas, e pretendem agora fazer suco de acerola, que será dividido igualmente entre os amigos durante o lanche da tarde.

Conhecendo o número de amigos, a quantidade de frutas colhidas, e sabendo que cada unidade da fruta é suficiente para produzir 50 ml de suco, escreva um programa para determinar qual o volume, em litros, que cada amigo poderá tomar.

### ENTRADA

O arquivo de entrada contém vários casos de teste. Cada caso de teste é composto por uma única linha, contendo dois números inteiros  $N$  e  $F$ , indicando respectivamente o número de amigos ( $1 \leq N \leq 1000$ ) e a quantidade de frutas colhidas ( $1 \leq F \leq 1000$ ). O final é indicado por uma linha que contém apenas dois zeros, separados por um espaço em branco.

### SAÍDA

Para cada caso de teste, seu programa deve imprimir uma única linha, contendo um número real, escrito com precisão de duas casas decimais, representando o volume do suco, em litros, a que cada amigo tem direito.

#### EXEMPLO DE ENTRADA

```
1 1
5 431
101 330
0 0
```

#### EXEMPLO DE SAÍDA

```
0.05
4.31
0.16
```

# Problema C: Olimpíadas

---

Tumbolia é um pequeno país ao leste da América do Sul (ou ao sul da América do Leste) que irá participar dos Jogos Olímpicos pela primeira vez na sua história. Apenas de sua delegação ser muito pequena comparada ao total de atletas que estarão em Pequim (as estimativas oficiais são de mais de dez mil atletas), a participação será fundamental para a imagem e para o turismo de Tumbólia.

Após selecionar os atletas, o Comitê Olímpico Tumboliano (COT) precisa comprar as passagens para eles. A fim de economizar dinheiro, o COT decidiu comprar apenas passagens da Air Rock. NO entanto, muitas das passagens da Air Rock já foram vendidas, uma vez que muitos tumbolianos desejam assistir aos Jogos. Sendo assim, o COT deverá comprar passagens de acordo com os assentos vagos em cada voo.

Todos os voos da Air Rock partem diariamente antes do meio-dia e chegam após o meio-dia; por isso, um atleta por tomar apenas um avião por dia. A Air Rock providenciou uma lista contendo todos os voos operados por ela e o número de assentos vagos em cada um (curiosamente, o número de assentos livres em um mesmo trecho é igual todos os dias).

O COT verificou que realmente é possível ir de Tumbólia para Pequim usando apenas voos da Air Rock mas, mesmo assim, o COT está tendo dificuldades para planejar a viagem de seus atletas. Por isso, o COT pediu para você escrever um programa que, dada a lista de voos da Air Rock, determina a menor quantidade de dias necessária para que todos os atletas cheguem em Pequim.

## ENTRADA

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém três inteiros  $N$ ,  $M$  e  $A$  indicando respectivamente a quantidade de aeroportos em que a Air Rock opera ( $2 \leq N \leq 50$ ), a quantidade de voos em que há assentos vagos ( $1 \leq M \leq 2450$ ) e quantos atletas a delegação tumboliana tem ( $1 \leq A \leq 50$ ).

Cada uma das  $M$  linhas seguintes contém uma descrição de voo com três inteiros  $O$ ,  $D$  e  $S$  que indicam respectivamente o aeroporto de origem ( $1 \leq O \leq N$ ), o aeroporto de destino ( $1 \leq D \leq N$  e  $O \neq D$ ) e a quantidade de assentos vagos naquele voo ( $1 \leq S \leq 50$ ). Os aeroportos são numerados de 1 a  $N$ ; o Aeroporto Internacional de Tumbólia é o aeroporto 1, e o Aeroporto Internacional de Pequim é o aeroporto  $N$ .

A existência de um voo de  $A$  para  $B$  **não** implica na existência de um voo de  $B$  para  $A$  (mas sempre há no máximo um voo de um aeroporto para outro em cada direção).

O final da entrada é indicado por  $N = M = A = 0$ .

## SAÍDA

Para cada caso de teste da entrada seu programa deve produzir uma linha contendo um inteiro, indicando a quantidade mínima de dias necessária para que todos os atletas tumbolianos cheguem em Pequim (alguns atletas podem chegar depois de outros, e eles não precisam chegar na mesma ordem em que partiram)

**EXEMPLO DE ENTRADA**

3 3 3  
1 2 2  
2 3 2  
1 3 1  
3 3 5  
1 2 1  
2 3 5  
3 1 4  
4 4 4  
1 4 1  
1 2 1  
2 3 1  
3 4 1  
0 0 0

**EXEMPLO DE SAÍDA**

2  
6  
3

## Problema D: Rápido com a Torta

---

A Pizzaria Pizazz é orgulhosa por entregar as pizzas aos clientes o mais rápido possível. Infelizmente, devido a contenção de gastos, eles só puderam contratar um motorista para as entregas. Ele espera por 1 ou mais (até 10) pedidos serem processados antes de iniciar as entregas. É desnecessário dizer que ele desejaria fazer a menor rota para entrega das mercadorias e retorno a pizzaria. Mesmo que isto implique em passar pela(s) mesma(s) localidades, ou mesmo pela pizzaria, mais de uma vez ao fazer o caminho. Você foi comissionado para fazer um programa para ajudá-lo.

### ENTRADA

A entrada consistirá em múltiplos casos de teste. A primeira linha irá conter um único inteiro  $n$  indicando o número de pedidos a entregar, onde  $1 \leq n \leq 10$ . Após esta, existirão  $n+1$  linhas, cada linha contendo  $n+1$  inteiros indicando o tempo de viagem entre a pizzaria (numerada como 0) e as demais localidades (números 1 até  $n$ ). Assim, o  $j$ -ésimo valor na  $i$ -ésima linha, indica o tempo para ir diretamente da localidade  $i$  até a localidade  $j$  sem visitar qualquer outra localidade no caminho. Note que pode haver caminhos mais rápidos entre  $i$  e  $j$  via outras localidades devido a diferentes limites de velocidades permitidas, distâncias, semáforos, etc. Adicionalmente, os tempos nem sempre são simétricos, isto é, o tempo de ir direto de  $i$  até  $j$  pode não ser o mesmo para ir de  $j$  até  $i$ . Um valor de entrada de  $n=0$  irá terminar as entradas.

### SAÍDA

Para cada caso de teste, você deverá ter como saída apenas um número, indicando o mínimo tempo para entregar todas as pizzas e retornar a pizzaria.

#### EXEMPLO DE ENTRADA

```
3
0 1 10 10
1 0 1 2
10 1 0 10
10 2 10 0
0
```

#### EXEMPLO DE SAÍDA

```
8
```

# Problema E: A Pizzaria do Bocão

---

Tradicionalmente após a nossa competição doméstica, os juízes e os competidores vão há uma pizzaria favorita, a Pizzaria do *Bocao*<sup>1</sup>. Os competidores estão realmente esfomeados após trabalharem arduamente durante cinco horas. Para conseguirem a sua pizza o mais rapidamente possível, eles então decidiram pedir a maior pizza para todos ao invés de várias pequenas. Contudo, eles se perguntaram se é possível colocar esta grande pizza retangular sobre a superfície de uma mesa redonda de tal que esta não ultrapasse a borda da mesa. Escreva um programa para ajudá-los!

## ENTRADA

O arquivo de entrada contém vários casos de teste. Cada caso de teste começa com um número inteiro  $r$ , o raio da superfície da mesa redonda na qual os competidores estão sentados. O término da entrada é dado por  $r = 0$ . Caso contrário,  $1 \leq r \leq 1000$ . Em seguida, tem 2 números inteiros  $l$  e  $w$  especificando a largura e comprimento da pizza,  $1 \leq w \leq l \leq 1000$ .

## SAÍDA

Para cada caso de teste, você deve indicar se a pizza pedida vai caber (*fits on the table.*) em cima da mesa ou não (*does not fit on the table.*). Siga o formato indicado no exemplo de saída. Uma pizza que apenas toca a borda da mesa sem interseção é considerada que esta cabe sobre a mesa, veja os 3 exemplos abaixo, que especificam o formato desejado.

### EXEMPLO DE ENTRADA

```
38 40 60
35 20 70
50 60 80
0
```

### EXEMPLO DE SAÍDA

```
Pizza 1 fits on the table.
Pizza 2 does not fit on the table.
Pizza 3 fits on the table.
```

<sup>1</sup> Embora a boca seja redonda, todos sabem que a pizza de lá é retangular!

# Problema F: Aritmética Primária

---

As crianças são ensinadas a somar números com vários dígitos a partir da direita para a esquerda, um dígito por vez. Muitos acham a operação “vai 1” (ou “carry”) – sendo esta feita a partir da posição de um dígito adicionado ao próximo – um desafio significativo. Seu trabalho é contar o número de operações “vai um” para cada conjunto de problemas de soma para que os educadores possam avaliar esta dificuldade.

## ENTRADA

Cada linha da entrada contém dois números inteiros sem sinal com menos de 10 dígitos. A última linha de entrada contém 0 0, que deve ser desconsiderada.

## SAÍDA

Para cada linha de entrada, exceto a última, deve-se calcular e imprimir o número de operações “vai um” que possam resultar da adição de dois números, no formato indicado abaixo.

### EXEMPLO DE ENTRADA

```
123 456
555 555
123 594
0 0
```

### EXEMPLO DE SAÍDA

```
No carry operation.
3 carry operations.
1 carry operation.
```

## Problema G: Número de Fitromacci

---

Uma sequência de números é dada pela seguinte relação de recorrência: os  $k$  primeiros números são iguais a 1; o  $n$ -ésimo ( $n > k$ ) valor é determinado pela soma dos  $k$  elementos anteriores.

Sua tarefa, neste exercício, será determinar o  $n$ -ésimo termo de Fitromacci. Por exemplo, para  $k = 3$  e  $n = 7$ , temos a sequência: 1 1 1 3 5 9 17 . . . e, portanto, o sétimo número neste caso é 17.

### ENTRADA

A primeira linha da entrada contém um inteiro  $T$  ( $1 \leq T \leq 1000$ ) indicando o número de instâncias. Cada instância é composta por uma linha contendo os inteiros  $k$  ( $1 \leq k \leq 7$ ), seguido por  $n$  ( $1 \leq n \leq 40$ ).

### SAÍDA

Para cada instância imprima, em uma linha, o  $n$ -ésimo termo da sequência. Você pode assumir que o número de Fitromacci é um inteiro de 64 bits (em C/C++, o tipo é "long long" e, em JAVA, "long").

### EXEMPLO DE ENTRADA

```
2
3 7
7 20
```

### EXEMPLO DE SAÍDA

```
17
24097
```