



Universidade Federal do Espírito Santo  
Departamento de Informática  
Programa de Educação Tutorial – PET EngComp  
E-mail: [petengcomp@inf.ufes.br](mailto:petengcomp@inf.ufes.br)  
Home-Page: [www.inf.ufes.br/~pet](http://www.inf.ufes.br/~pet)  
Tel. (27) 3335-2161

---

# Topcom 4

Torneio de programação de computadores

## PROVA

Realização:



Apoio:



## PROBLEMA A

### Contando Retângulos

Em alguns problemas aparece a dificuldade de encontrar o número de retângulos (ou círculos, ou triângulos, ...) de diferentes tamanhos na figura. Nós consideramos o problema de contar retângulos (incluindo quadrados) numa placa retangular.

Dado uma placa retangular com  $n$  linhas e  $m$  colunas, com posições válidas marcadas com 1 e posições não válidas marcadas com 0, nós queremos contar o número de retângulos (incluindo quadrados) na placa formada por quadrados marcados com 1.

#### A ENTRADA

A entrada constará de uma série de problemas, com cada problema em série de linhas. Nas primeira e segunda linhas, as linhas ( $n$ ) e as colunas ( $m$ ) da placa indicada, e, nas próximas  $n$  linhas a placa na representação, com uma linha da placa em cada linha, e  $m$  0s ou 1s (sem espaços) em cada linha. Quando a entrada de cada problema termina, o próximo problema começa na linha seguinte. A entrada termina quando 0 aparece como dimensão da placa. Ambas dimensões da placa são menores que 100.

#### A SAÍDA

A solução de cada problema aparece na linha, sem separação entre linhas. Por exemplo, na placa:

```
11
01
```

os retângulos são:

```
1-
--
```

```
-1
--
```

```
--
-1
```

```
11
--
```

```
-1
```

-1

**EXEMPLO DE ENTRADAS**

2  
2  
11  
01  
0

**EXEMPLO DE SAÍDA**

5

## **PROBLEMA B**

### **Números de Fibonacci**

Uma sequência de Fibonacci é calculada através da soma de dois membros anteriores, sendo que os dois primeiros termos valem 1.

$$f(1) = 1, f(2) = 1, f(n > 2) = f(n - 1) + f(n - 2)$$

#### **A ENTRADA E A SAÍDA**

Sua tarefa é tomar números como entrada (um por linha) e imprimir o número Fibonacci correspondente. Nenhum número fibonacci com mais de 10 dígitos estará na massa de testes. Por exemplo,  $f(20) = 6765$  tem 4 dígitos.

#### **EXEMPLO DE ENTRADA**

3  
45

#### **EXEMPLO DE SAÍDA**

2  
1134903170

## PROBLEMA C

### Bin Packing Ecológico

*Bin packing*, ou o arranjo de objetos de certos pesos em diferentes caixas sujeitas a certas restrições, é, historicamente, um problema interessante. Alguns problemas *bin packing* são NP-completos, entretanto são passíveis de solução através de programação dinâmica ou de aproximações de soluções heurísticas.

Neste problema você resolverá um problema *bin packing* que lida com reciclagem de vidro.

Reciclar vidro requer que o vidro seja separado por uma cor entre três categorias: vidro marrom, vidro verde e vidro transparente. Neste problema você receberá três caixas de reciclagem, cada uma contendo um número específico de garrafas marrons, verdes e transparentes. Para que sejam recicladas, as garrafas deverão ser movidas de tal forma que cada caixa contenha garrafas de apenas uma cor.

O problema é minimizar o número de garrafas que deverão ser movidas. Você pode assumir que o único problema é minimizar o número de movimentos entre caixas.

Para os propósitos deste problema, cada caixa possui capacidade infinita e a única restrição é mover as garrafas de tal forma que cada caixa possua garrafas de apenas uma cor. O número total de garrafas nunca será maior que  $2^{31}$ .

#### A ENTRADA

A entrada consiste de uma série de linhas com cada linha contendo 9 inteiros. Os primeiros três inteiros de uma linha representam o número de garrafas marrons, verdes e transparentes (respectivamente) na caixa número 1, os três seguintes representam o número de garrafas marrons, verdes e transparentes (respectivamente) na caixa número 2 e os três últimos representam o número de garrafas marrons, verdes e transparentes (respectivamente) na caixa número 3. Por exemplo, a linha 10 15 20 30 12 8 15 8 31 indica que há 20 garrafas transparentes na caixa 1, 12 garrafas verdes na caixa 2 e 15 garrafas marrons na caixa 3.

Inteiros em uma linha serão separados por um ou mais espaços em branco. Seu programa deve processar todas as linhas do arquivo de entrada.

#### A SAÍDA

Para cada linha da entrada, existirá uma linha na saída indicando qual cor vai em cada caixa, de maneira a minimizar o número de movimentos de garrafas. Você deve imprimir também o número mínimo de movimentos de garrafas.

4º Torneio de Programação – TOPCOM 4  
PET – Engenharia de Computação

A saída deve consistir de uma string de três caracteres maiúsculos ‘G’, ‘B’, ‘C’ (representando, respectivamente, as cores verde, marrom e transparente) representando a cor associada a cada caixa.

O primeiro caracter da string representa a cor associada com a primeira caixa, o segundo caracter da string representa a cor associada com a segunda caixa, e o terceiro caracter da string representa a cor associada com a terceira caixa.

O inteiro indicando o número mínimo de movimentos de garrafa deve ser impresso após a string.

Se houver mais de uma configuração de cores para as caixas que minimizem os movimentos de garrafa, então a primeira string, de acordo com a ordem alfabética, deve ser impressa como solução.

**EXEMPLO DE ENTRADA**

```
1 2 3 4 5 6 7 8 9
5 10 5 20 10 5 10 20 10
```

**EXEMPLO DE SAÍDA**

```
BCG 30
CBG 50
```

## PROBLEMA D

### Crise de Energia

Durante a crise de energia na Nova Zelândia um esquema de contingência foi desenvolvido para desligar a energia de áreas do país de uma maneira sistemática e totalmente justa. O país foi dividido em  $N$  regiões (Auckland foi a região número 1 e Wellington 13). Um número,  $m$ , seria escolhido “aleatoriamente”, e a energia seria primeiramente desligada na região 1 e então em cada  $m^a$  região depois dela indo de 1 até  $N$  e ignorando regiões que já tiveram sua energia desligada. Por exemplo, se  $N = 17$  e  $m = 5$ , a energia seria desligada nas regiões em ordem: 1,6,11,16,5,12,2,9,17,10,4,15,14,3,8,13,7.

O problema é que é claramente mais justo desligar a energia de Wellington por último (já que é lá que estão as estações geradoras de energia). Então para um dado  $N$ , o número “aleatório”  $m$  deve ser cuidadosamente escolhido de modo que a região 13 seja a última selecionada.

Escreva um programa que lerá o número de regiões e então determine o menor número  $m$  que assegure que Wellington (região 13) possa funcionar enquanto o resto do país é desligado.

#### A ENTRADA E A SAÍDA

A entrada consistirá numa série de linhas, cada uma contendo o número de regiões ( $N$ ) com  $13 \leq N < 100$ . O arquivo deverá ser terminado com uma linha contendo um simples 0.

A saída consistirá numa série de linhas, uma para cada linha da entrada. Cada linha consistirá no número  $m$  conforme explicação acima.

#### EXEMPLO DE ENTRADA

```
17
0
```

#### EXEMPLO DE SAÍDA

```
7
```

## PROBLEMA E

### Interceptação

Você está pra escrever um programa que tem que decidir se um dado segmento de reta intercepta um dado retângulo.

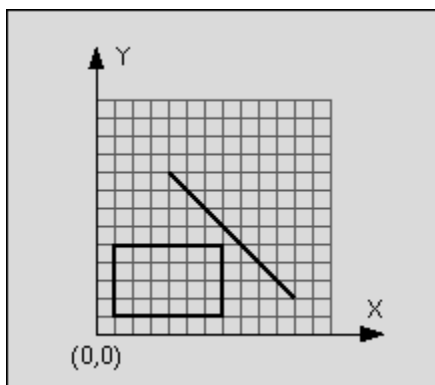
#### *Um exemplo:*

reta:                      ponto de partida:                      (4, 9)

                                 ponto final:                      (11, 2)

retângulo:              canto esquerdo superior:                      (1, 5)

                                 canto direito inferior:                      (7, 1)



**Figura:** Segmento de reta não intercepta o retângulo

A reta intersesta o retângulo se ela e o retângulo possuem pelo menos um ponto em comum. O retângulo consiste de quatro segmentos de retas e a área interior. Apesar de todos os valores de entrada serem números inteiros, pontos válidos de interceptação não devem necessariamente estar sobre a grade de números inteiros.

#### **A ENTRADA**

A entrada consiste em n casos de teste. A primeira linha do arquivo de entrada contém o



4º Torneio de Programação – TOPCOM 4  
PET – Engenharia de Computação

número  $n$ . Cada linha seguinte contém um caso de teste no formato:

$xstart$   $ystart$   $xend$   $yend$   $xleft$   $ytop$   $xright$   $ybottom$

onde  $(xstart, ystart)$  é o ponto inicial e  $(xend, yend)$  o final do segmento de reta e  $(xleft, ytop)$  o ponto superior esquerdo e  $(xright, ybottom)$  o inferior direito do retângulo. Os oito números são separados por espaço.

### A SAÍDA

Para cada caso de teste no arquivo de entrada, o arquivo de saída deve conter uma linha contendo uma letra “T” se o segmento de reta intercepta o retângulo ou “F” se não.

### EXEMPLO DE ENTRADA

```
1
4 9 11 2 1 5 7 1
```

### EXEMPLO DE SAÍDA

```
F
```

## **PROBLEMA F**

### **Pilha de Pedras**

Você tem  $n$  pedras de pesos conhecidos  $W_1 \dots W_n$ . Escreva um programa que rearranjará as pedras em dois montes tais que a diferença entre os pesos dos montes seja mínima.

#### **A ENTRADA**

A entrada contém conjuntos de testes com o número de pedras  $N$  ( $1 \leq N \leq 20$ ) e pesos das pedras  $W_1 \dots W_n$  ( $1 \leq W_i \leq 100000$ ) delimitados por '\n'. Um teste é seguido diretamente pelo outro.

#### **A SAÍDA**

Seu programa deve gerar um número como saída, representando a diferença de peso mínima possível entre os montes de pedras.

#### **EXEMPLO DE ENTRADA**

```
5
5
8
13
27
14
```

#### **EXEMPLO DE SAÍDA**

```
3
```

## **PROBLEMA G**

### **Produto de Dígitos**

Sua tarefa é achar o menor número inteiro positivo  $Q$  de modo que o produto dos dígitos de  $Q$  seja exatamente igual a  $N$ .

#### **A ENTRADA**

A entrada contém o número inteiro  $N$  ( $0 \leq N \leq 10^9$ ). Um teste é seguido diretamente do outro.

#### **A SAÍDA**

Seu programa deve imprimir na saída somente o número  $Q$ . Se não existir tal número, imprima -1.

#### **EXEMPLO DE ENTRADA**

10  
20

#### **EXEMPLO DE SAÍDA**

25  
45

## PROBLEMA H

### Coloração de Mapas

Considere um mapa geográfico com  $N$  países numerados de 1 a  $N$  ( $0 < N < 99$ ). Para cada país são conhecidos o número de países, diferentes dele mesmo, que estão conectados à sua borda. De cada país é possível alcançar qualquer um outro, eventualmente atravessando algumas bordas. Escreva um programa que determina se é possível colorir o mapa com somente duas cores (vermelha e azul) de tal forma que se dois países estão conectados suas cores devem ser diferentes. A cor do primeiro país é vermelha. Seu programa deve gerar como saída uma possível coloração para os demais países ou mostrar que tal coloração é impossível.

#### A ENTRADA

Na primeira linha está escrito o número  $N$ . Nas  $N$  linhas seguintes, a  $I$ -ésima linha contém os países com os quais o  $I$ -ésimo país está conectado. Cada inteiro nesta linha é maior que  $I$ , exceto o último, que é zero e indica que não há mais países conectados ao país  $I$ . Se uma linha contém apenas 0, significa que o  $I$ -ésimo país não está conectado com nenhum outro cujo número é maior que  $I$ .

#### A SAÍDA

A saída contém exatamente uma linha. Se a coloração é possível, esta linha deve conter uma lista de zeros e uns, sem nenhum separador entre eles. O  $I$ -ésimo dígito nesta sequência é a cor do  $I$ -ésimo país. Zero corresponde à cor vermelha, e um corresponde à cor azul. Se a coloração é impossível, deverá ser impresso o inteiro -1.

#### EXEMPLO DE ENTRADA

```
3
2 0
3 0
0
```

#### EXEMPLO DE SAÍDA

```
010
```