



## **XI Maratona Interna de Programação**

**29 de abril de 2017**

### **Caderno de Problemas**

- Este caderno contém seis problemas, com páginas numeradas de 1 a 8, descontando a primeira.
- Verifique se o caderno está completo.

### **Informações Gerais**

#### **A) Sobre o arquivo de solução e submissão**

1. O arquivo de solução deve ter o mesmo nome que o especificado no enunciado (logo após o título).
2. Antes de submeter a sua solução, confirme se escolheu a linguagem correta e está com o nome do arquivo correto.

#### **B) Sobre a entrada**

1. A entrada de seu programa deve ser lida da entrada padrão (não use interface gráfica).
2. A entrada é composta por vários casos de teste, especificados no enunciado.
3. Em alguns problemas, o final da entrada coincide com o final do arquivo. Em outros, o final é especificado por alguma combinação de caracteres ou dígitos.

#### **C) Sobre a saída**

1. A saída de seu programa deve ser escrita na saída padrão.
2. Não exiba qualquer mensagem não especificada no enunciado.

## Problema A

### A257299

Source file: a257299.{c | cpp | java | py}

Author: Erich Rodrigues Ferrares

A sequência **A257299** da OEIS (*The Online Encyclopedia of Integer Sequences*) é formada por números **N** para os quais cada um dos dígitos 0-9 aparece exatamente uma vez como primeiro dígito na órbita de  $n$  sob iterações de  $n \rightarrow (\text{primeiro dígito de } n) * (n \text{ com o primeiro dígito removido})$  até que um número com apenas um dígito seja alcançado; Zeros à esquerda não são permitidos.

Um exemplo de número presente nessa sequência é o **9848**, pois  $9848 \rightarrow 9 * 848 = 7632 \rightarrow 7 * 632 = 4424 \rightarrow 4 * 424 = 1696 \rightarrow 1 * 696 = 696 \rightarrow 6 * 96 = 576 \rightarrow 5 * 76 = 380 \rightarrow 3 * 80 = 240 \rightarrow 2 * 40 = 80 \rightarrow 8 * 0 = 0$ , cada um dos dígitos 0-9 aparecem exatamente uma vez como o primeiro dígito.

Outros exemplos são os números 56648, 68648, 77712, 84157, 87207, 98142.

A sua tarefa neste exercício será dizer se um número **N** está presente ou não na sequência **A257299**.

#### Entrada

A primeira linha da entrada contém um número **T**, correspondente ao número de casos de teste. Segue-se **T** linhas, cada um contendo um número **N**.

#### Saída

A saída deve conter a letra 'T', caso o número **N** pertença a sequência **A257299**, senão, deve apresentar a letra 'F'.

#### Exemplo

5	
9848	T
9849	F
123456789	F
257299	F
2259492	T

## Problema B

# Anti-Dependências Perpendiculares

*Source file:* dp.{c | cpp | java | py }

*Author:* Sérgio Hilgert Junior

Recentemente um professor da faculdade Cetaf descobriu que muitos alunos estão ficando com Dependências Perpendiculares em algumas matérias. Ao longo de três anos o professor Giulio Bertolomeu realizou uma pesquisa avançada e descobriu que essas Dependências Perpendiculares são indiretamente proporcionais a quantidade dos diferentes divisores das horas que um aluno frequentou a matéria. Por exemplo se um aluno frequentou a matéria X por 12 horas então ele terá 6 pontos de Anti-DP, pois 12 possui os divisores 1, 2, 3, 4, 6 e 12.

Giulio quer investir em sua pesquisa mas não tem tempo, pois está orientando muitos alunos. Sua tarefa é dado uma determinada quantidade de horas descobrir o número de pontos de Anti-Dependências Perpendiculares que aquele aluno possui.

### Entrada

Um inteiro  $N$  ( $1 \leq N < 9 \times 10^{18}$ ).

### Saída

A quantidade de pontos de Anti-DP daquele aluno (diferentes divisores que aquele número possui).

### Exemplo

1	1
2	2
4	3
6	4
12	6
5040	60

## Problema C

### O caso da cobra

*Source file:* snake\_case.{c | cpp | java | py}

*Author:* Lucas Narciso

Giréso ingressou na Cetaf para estudar Desenvolvimento e Análise de Sistemas. Ele estudou e participou das maratonas de programação e logo conseguiu um bom estágio. Na empresa em que ele entrou, é usada a linguagem Ruby. Seguindo a convenção da comunidade Ruby, os nomes das variáveis são escritos no formato **snake\_case** e os nomes das classes em **UpperCamelCase**. Giréso, entretanto, só havia estudado a linguagem Java até então e estava acostumado a nomear as suas variáveis no formato **lowerCamelCase**. Ele já escrevia o nome das classes no formato **UpperCamelCase**, mas, tendo ficado muito nervoso no seu primeiro projeto na empresa, se confundiu todo e acabou escrevendo o nome de suas classes em **snake\_case** e o nome das suas variáveis em **lowerCamelCase**. Para que seu chefe não ache que tomou uma decisão ruim ao contratá-lo, ele pediu a sua ajuda.

#### Entrada

Cada linha da entrada contém ou uma palavra no formato **camelCase**, representando uma variável nomeada incorretamente, ou a palavra “class” (sem aspas) seguida de uma palavra no formato **snake\_case**, representando uma classe nomeada incorretamente. Nenhuma linha da entrada conterá mais do que **160** caracteres. A entrada termina com EOF.

#### Saída

Para cada linha da entrada, imprima a sua correção para o *case* correto, ou seja, se a linha representar o nome de uma variável, imprima uma linha com esse nome em **snake\_case**; se a linha representar o nome de uma classe, imprima uma linha com “class” (sem aspas) e o nome da classe em **UpperCamelCase**, separados por um espaço em branco.

#### Exemplo

class user currentUser class amazing_pencil_sharpener grandma theStruggleIsReal	class User current_user class AmazingPencilSharpener grandma the_struggle_is_real
---	---

## Problema D

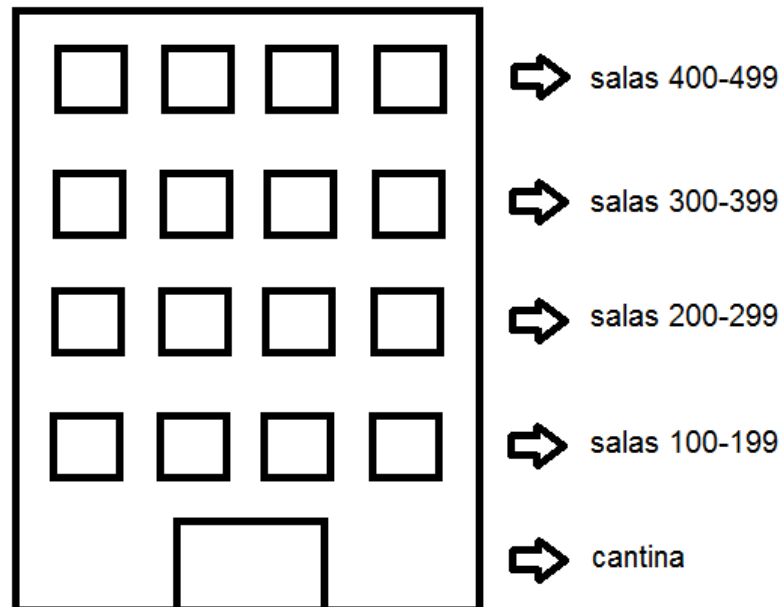
### Preguiça de Escada

Source file: preguica.{c | cpp | java | py}

Author: Heitor Miranda Moreira

Lucérgio é um estudante muito preguiçoso. Ele acabou de concluir o Ensino Médio e ficou muito feliz por ter passado no vestibular do curso de Análise e Desenvolvimento de Sistemas da faculdade Cetaf.

Ao chegar no prédio da faculdade pela primeira vez, Lucérgio ficou assustado com a quantidade de andares. Por ser muito preguiçoso, ele cogitou até mesmo a desistir do curso quando soube que teria que ficar subindo e descendo todas aquelas escadas todos os dias, mas seus pais o incentivaram a continuar. O desenho abaixo representa o prédio da faculdade e a localização das salas em cada andar.



O curso de Análise e Desenvolvimento de Sistemas na faculdade Cetaf tem muitos alunos, então a coordenação o divide em duas turmas: turma A e turma B. Ambas as turmas têm a mesma grade escolar, com quatro aulas às segundas, quartas e sextas-feiras e seis aulas às terças e quintas-feiras, porém as aulas são distribuídas em salas diferentes pelo prédio. Todos os dias, o intervalo sempre é realizado após as duas primeiras aulas. A tabela abaixo representa a sala onde as duas turmas têm aulas durante todos os dias da semana. Os números à esquerda representam a turma A, enquanto os números à direita representam a turma B.

Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
108 / 107	303 / 410	101 / 306	401 / 107	412 / 303
108 / 107	303 / 410	101 / 306	401 / 107	412 / 303
INTERVALO	INTERVALO	INTERVALO	INTERVALO	INTERVALO
412 / 306	401 / 212	207 / 103	101 / 103	303 / 212
412 / 306	401 / 212	207 / 103	101 / 103	303 / 212
	207 / 303		108 / 410	
	207 / 303		108 / 410	

Lucérrio, como não é bobo, quer escolher a turma que tem os horários e salas de aula organizados de forma que ele precise utilizar as escadas a menor quantidade de vezes possível. Assim, ele quer analisar os horários das duas turmas e escolher a melhor opção delas, considerando o fato de que, às terças e quintas-feiras, ele terá de descer ao térreo para comprar lanche na cantina e então retornar à sala onde terá a próxima aula. Além disso, todos os dias, após as aulas, ele tem que descer até o térreo para ir embora para casa.

Como ele é muito preguiçoso, pediu a sua ajuda para fazer um programa que calcule qual é a melhor turma para ele. Assim, ele pode usá-lo durante toda a sua graduação, podendo trocar as turmas a cada semestre que passa.

### Entrada

A primeira linha da entrada consiste em um inteiro  $N$  ( $1 \leq N \leq 100$ ), indicando o número de casos de teste. Cada caso de teste contém dez linhas: cinco para a turma A e cinco para turma B. Cada uma dessas linhas representa um dia da semana, de segunda a sexta-feira e contém o número das salas  $S$  ( $100 \leq S \leq 499$ ) onde as aulas são realizadas.

### Saída

A saída para cada caso de teste deve ser uma linha contendo uma letra, “A” ou “B”, indicando qual turma é a melhor opção para Lucérrio, ou “tanto faz”, caso as duas opções sejam iguais.

### Exemplo

2 108 108 412 412 303 303 401 401 207 207 101 101 207 207 401 401 101 101 108 108 412 412 303 303 107 107 306 306 410 410 212 212 303 303 306 306 103 103 107 107 103 103 410 410	B tanto faz
--	----------------

303 303 212 212 101 101 204 204 411 411 303 303 101 101 204 204 101 101 101 101 106 106 302 302 101 101 405 405 402 402 401 401 303 303 104 104 211 211 211 211 307 307 110 110 110 110 307 307 204 204 303 303	
---	--

## Problema E

# TresTresTres

*Source file:* tres.{c | cpp | java | py}

*Author:* Sérgio Hilgert Junior

Leumas tem um grave problema com número três, quando ele vê um número três ele não consegue se controlar e começa a programar em C++.

Leumas quer saber, dado um determinado número **N**, quantos números de 0 até **N** não possuem o dígito três. Por exemplo: o número 10, possui apenas um antecessor com um dígito três, então o resultado seria 9. O número 45 tem os antecessores 3, 13, 23, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 43 que contém o número três, então o resultado seria 31.

Sua tarefa é dado um número **N**, descobrir quantos números de 0 até **N** (inclusive) não possuem o dígito 3.

### Entrada

Um inteiro **N** ( $0 \leq N \leq 1000000$ ).

### Saída

Um inteiro representando a quantidade de números que não possuem o dígito 3.

### Exemplo

10	9
45	31
578	385



## Problema F

### Votos

*Source file:* votos.{c | cpp | java | py}

*Author:* Lucas Narciso

O governo do estado do Lago de Fevereiro teve alguns problemas técnicos recentemente e perdeu alguns dados da última eleição. Ela tem os nomes dos três candidatos mais votados, quantos se candidataram e uma lista dos votos de cada cidade, mas perdeu quantos votos cada um dos três primeiros recebeu. Ajude o governo do Lago de Fevereiro a recuperar os dados das eleições!

Dados os números de candidatos de várias cidades e uma lista dos votos, mostre quais são as três maiores quantidades de votos.

#### Entrada

A entrada consiste de um número **T**, representando quantidade de cidades que precisa das informações. Segue a descrição das **T** cidades. Cada descrição contém duas linhas: uma linha com um único inteiro **N** ( $3 \leq N \leq 10000$ ), indicando o número de candidatos daquela cidade; e uma linha com **N** inteiros **i** ( $0 \leq i \leq 1000000$ ), que são os votos apurados.

#### Saída

Para cada cidade, imprima uma linha com três inteiros **em ordem decrescente**, indicando as três maiores quantidades de votos que ela possuiu.

#### Exemplo

4	7 6 5
7	4096 2048 1024
1 2 3 4 5 6 7	10 3 2
8	1000 11 11
256 16 1024 512 128 4096 64 2048	
5	
1 2 3 10 2	
10	
10 10 10 10 10 1000 10 11 11 11	