



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

Topcom 5

Torneio de Programação de Computadores

WARM UP

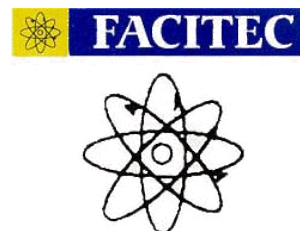
Realização:



Apoio:



Patrocínio:



5º Torneio de Programação – TOPCOM 5
PET – Engenharia de Computação - UFES

PROBLEMA A

AEROPORTOS

5º Torneio de Programação – TOPCOM 5

A crescente utilização do transporte aéreo por especialistas, que prevêem que o congestionamento em aeroportos poderá se tornar um grande problema no futuro. Os números atuais já são alarmantes: relatórios oficiais demonstram que na Europa, em junho de 2001, houve uma média de 7.000 atrasos de vôos por dia. Preocupada com a previsão dos seus especialistas em tráfego aéreo, a Associação de Transporte Aéreo Internacional (ATAI) está começando um estudo para descobrir quais são os aeroportos onde o tráfego aéreo pode vir a ser mais problemático no futuro.

Como programador recém contratado pela ATAI você foi encarregado de escrever um programa para determinar, a partir de uma listagem de aeroportos e vôos, qual aeroporto possui maior probabilidade de congestionamento no futuro. Como medida da probabilidade de congestionamento, será utilizado neste estudo o número total de vôos que chegam ou que partem de cada aeroporto.

A ENTRADA

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém dois números inteiros A e V, que indicam respectivamente o número de aeroportos e o número de vôos. Os aeroportos são identificados por inteiros de 1 a A. As V linhas seguintes contém cada uma a informação de um vôo, representada por um par de números inteiros positivos X e Y, indicando que há um vôo do aeroporto X para o aeroporto Y. O final da entrada é indicado quando $A = V = 0$.

Restrições:

$0 \leq A \leq 100$ (A = 0 apenas para indicar o fim da entrada)

$0 \leq V \leq 10000$ (V = 0 apenas para indicar o fim da entrada)

$1 \leq X \leq A$

$1 \leq Y \leq A$

$X \neq Y$

A SAÍDA

Para cada conjunto de teste da entrada seu programa deve produzir três linhas. A primeira linha identifica o conjunto de teste, no formato "Teste n", onde n é numerado a partir de 1. A segunda linha deve conter o identificador do aeroporto que possui maior tráfego aéreo. Caso mais de um aeroporto possua este valor máximo, você deve listar todos estes aeroportos, em ordem crescente de identificação, e separados por pelo menos um espaço em branco. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

EXEMPLO DE ENTRADA

```
5 7
1 3
2 1
3 2
3 4
4 5
3 5
2 5
3 5
1 3
1 2
3 2
1 2
```

PROBLEMA B

FUNÇÃO DE ACKERMANN

Uma função de Ackermann tem a característica de que o comprimento da sequência de números gerados pela função não pode ser computada diretamente do valor de entrada. Especificamente, uma função de Ackermann de inteiros é a seguinte:

$$X_{n+1} = \begin{cases} \frac{X_n}{2}, & \text{se } X_n \text{ é par} \\ 3X_n + 1, & \text{se } X_n \text{ é ímpar} \end{cases}$$

Esta Ackermann tem a característica de eventualmente convergir para 1. Alguns exemplos seguem, nos quais o valor inicial é mostrado em colchetes, seguido pela sequência de valores gerados, seguido pelo comprimento da sequência entre chaves:

```
[10] 5 16 8 4 2 1 {6}
[13] 40 20 10 5 16 8 4 2 1 {9}
[14] 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 {17}
[19] 58 29 88 44 22 ... 2 1 {20}
[32] 16 8 4 2 1 {5}
[1] 4 2 1 {3}
```

A ENTRADA E A SAÍDA

Seu programa deve ler como entrada uma série de pares de valores que representam o primeiro e o último número de domínio de valores. Para cada domínio, determine qual valor irá gerar a série de Ackermann mais longa antes que ela atinja o valor 1. O último par de valores será 0, 0. A saída do seu programa deve ser como segue:

Entre L e H, V gera a mais longa sequência de S valores. Em que:

L = menor valor do domínio.

H = maior valor do domínio.

V = primeiro valor que gera a mais longa sequência.

S = comprimento da sequência gerada.

Se dois ou mais valores geram a mais longa sequência, será mostrado somente o menor valor.

EXEMPLO DE ENTRADA

```
1 20
35 55
0 0
```

EXEMPLO DE SAÍDA

```
Entre 1 e 20, 18 gera a mais longa sequência de 20 valores.
Entre 35 e 55, 54 gera a mais longa sequência de 112 valores.
```

PROBLEMA C

MÉTODO DE CRIPTOGRAFIA DIVERTIDO

Um estudante da UFES brinca com um novo método de criptografia. Este método consiste dos seguintes passos.

1. Leia um número M para criptografar: $M = 265$
2. Interprete M como um número decimal: $X1 = 265$ (decimal)
3. Converta a representação decimal de M para a representação: $X1 = 100001001$ (binário)
4. Seja $b1$ igual ao número de 1's na representação binária: $b1 = 3$
5. Interprete M como um número hexadecimal: $X2 = 265$ (hexadecimal)
6. Converta a representação hexadecimal de M para sua representação binária: $X2 = 1001100101$
7. Seja $b2$ igual ao número de 1's na representação binária anterior: $b2 = 5$
8. A criptografia é o resultado de M xor $(b1*b2)$: M xor $(3*5) = 262$

Sua tarefa é escrever um programa que lê um número e imprime como resultado os números $b1$ e $b2$.

Dica: 265 (hexadecimal) = $2 \times 16^2 + 6 \times 16^1 + 5 \times 16^0 = 613$ (decimal)

A ENTRADA

A primeira linha conterá o número N que é o número de casos que você deve processar. Cada uma das N linhas ($0 < N \leq 1000$) conterá um número M ($0 < M \leq 9999$, na representação decimal) que é o número que você quer criptografar.

A SAÍDA

Você deve imprimir N linhas, cada uma contendo o número $b1$ e $b2$, nesta ordem, separado por um espaço simples.

EXEMPLO DE ENTRADA

```
2
265
111
```

EXEMPLO DE SAÍDA

```
3 5
6 3
```

PROBLEMA D

PRA SAIR DO ZERO!

Calcular o resto da divisão inteira de A por B.

A ENTRADA

Um número qualquer de linhas de teste contendo os números A e B. Uma linha contendo 0 0 termina a entrada.

A SAÍDA

Para cada par A e B, imprimir uma linha contendo o resto da divisão inteira de A por B.

EXEMPLO DE ENTRADA

```
7 5
6 3
0 0
```

EXEMPLO DE SAÍDA

```
2
0
```