

ToPAS'05

Torneio de Programação para Alunos do Secundário

Departamento de Ciência de Computadores

<http://www.dcc.fc.up.pt/topas/>

Conjunto de Problemas



Faculdade de Ciências da Universidade do Porto

14 de Maio de 2005

Este conjunto de problemas deverá conter sete (7) problemas e dezoito (18) páginas.
Se faltar algum problema, por favor avise a organização.

ToPAS'05

Torneio de Programação para Alunos do Secundário

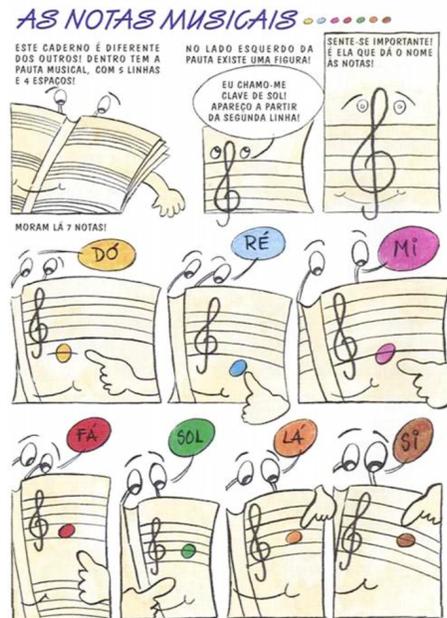
Dep. Ciência de Computadores – FCUP
14 de Maio de 2005

Contents

Problema A: Música, Maestro!	3
Problema B: Reunião	5
Problema C: Matrículas	7
Problema D: Triângulo de Sierpinski	9
Problema E: Telepreso	11
Problema F: Relógio Digital	15
Problema G: Sentar ou não Sentar?	17

Problema A

Música, Maestro!



Problema

Na música ocidental, encontramos sete notas musicais, ordenadas por alturas crescentes:

DÓ	RÉ	MI	FÁ	SOL	LÁ	SI	DÓ	RÉ	MI	...
1	2	3	4	5	6	7	1	2	3	...

Por definição, um intervalo musical é a diferença que existe entre duas notas. Como temos sete notas musicais, e tendo em conta que a mesma nota pode formar um intervalo de oitava consigo própria, encontramos também sete intervalos musicais.

Dependendo do número de graus que existe entre a primeira e a segunda nota do intervalo, este pode ser classificado como de segunda, terceira, quarta, quinta, sexta, sétima e finalmente de oitava. Consideremos, por exemplo, as notas de lá a dó. Seguindo a sequência dos graus, lá, si e dó, contamos três graus e concluímos tratar-se de um intervalo de terceira.

Tarefa

Escrever um programa que dadas duas notas musicais, identificadas pelo número respectivo, calcule o número de graus entre elas e determine o intervalo musical.

Input

A primeira linha de um input é a primeira nota musical, a segunda linha de um input é a segunda nota musical.

Output

O output será uma linha que apresenta o nome do intervalo.

Exemplo 1: Input

2
4

Exemplo 1: Output

intervalo de terceira

Exemplo 2: Input

4
2

Exemplo 2: Output

intervalo de sexta

Exemplo 3: Input

1
7

Exemplo 3: Output

intervalo de setima

Problema B

Reunião

Problema

“Dans l’Océan Indien, une petite terre française s’emploie à bousculer les clichés habituels des îles tropicales. Des cocotiers et des plages, il y en a, bien sûr, à la Réunion. L’eau turquoise du lagon, l’ourlet blanc de la barrière de corail sont au rendez-vous des rêves exotiques.”

Dizem que na ilha da Reunião não é fácil marcar reuniões. Há muitos outros afazeres... Para agradar a uns e outros vai-se tentando saber qual a disponibilidade de cada um. No fim, espera-se poder fixar uma hora num intervalo de comum acordo, e como, no meio do intervalo está a virtude, será tal a hora da reunião.

Tarefa

Faça um programa que dado o um conjunto de intervalos $[a, b]$, com extremos inteiros a e b entre 8 e 18, fixará como hora da reunião o ponto médio do intervalo comum, ou indicará, se é *impossível*.

Input

É dado um número n de pessoas envolvidas e nas linhas seguintes pares a, b , separados por espaço, que indicam as preferências das n pessoas.

Output

O output será “impossível” se não existir nenhuma hora possível ou “x” ou então “x e meia”, sendo x um inteiro.

Exemplo 1: input

```
3
10 15
8 12
11 12
```

Exemplo 1: output

```
11 e meia
```

Exemplo 2: input

4
10 15
8 12
11 12
17 19

Exemplo 2: output

impossivel

Exemplo 3: input

2
10 15
15 16

Exemplo 3: output

15

Problema C

Matrículas

Problema

No reino da Lusitânia, o Centro do Automóvel e das Rodovias (CAR) apresentou recentemente um novo plano para a atribuição de matrículas automóveis. Esse plano, para além de propor um novo modelo de atribuição, apresenta um elaborado estudo estatístico sobre os anos passados. Na elaboração desse estudo, uma das tarefas mais relevantes foi calcular o número de veículos matriculados durante os últimos anos.

Tarefa

Escrever um programa que dadas as matrículas do primeiro e do último veículo automóvel matriculado num dado ano, calcule o número de veículos matriculados nesse mesmo ano. Considere apenas matrículas do tipo AA-AA-LL, em que A é um algarismo e L é uma das 23 letras do alfabeto português. A primeira matrícula é 00-00-AA e a última é 99-99-ZZ. A atribuição de matrículas segue o seguinte critério: (i) partindo da última matrícula atribuída, o segundo par de algarismos é incrementado em uma unidade; (ii) se o segundo par de algarismos atingir o valor 99, este passa a 00 e o primeiro par de algarismos é incrementado em uma unidade; (iii) se o primeiro par de algarismos atingir o valor 99, este passa a 00 e a segunda letra é incrementada em uma unidade, segundo a ordem das letras no alfabeto; (iv) se a segunda letra atingir o valor Z, esta passa a A e a primeira letra é incrementada em uma unidade.

Input

A primeira linha de input é um inteiro N (≤ 100) que indica o número de pares de matrículas a considerar. As N linhas seguintes são pares de matrículas do tipo AA-AA-LL, em que A é um algarismo e L é uma das 23 letras do alfabeto português. A primeira matrícula de cada par diz respeito ao primeiro veículo automóvel matriculado num dado ano, enquanto a segunda diz respeito ao último veículo matriculado nesse mesmo ano.

Output

O output é um conjunto de N linhas, em que cada linha apresenta o número de veículos matriculados tendo por base o par de matrículas indicado.

Exemplo de Input

3

00-00-AA 99-99-AA

00-00-AZ 00-00-BA

01-99-XZ 99-01-ZV

Exemplo de Output

10000

10001

449703

Problema D

Triângulo de Sierpinski

Problema

O triângulo de Pascal é uma representação possível para os coeficientes da expansão do binómio de Newton: $(a + b)^{n+1}$. O triângulo pode ser definido usando a regra: o elemento de ordem $k + 1$ na linha $i + 1$ resulta da soma dos elementos de ordem k e $k + 1$ da linha i . O triângulo para $n = 8$ é:

```
1:          1
2:         1 1
3:        1 2 1
4:       1 3 3 1
5:      1 4 6 4 1
6:     1 5 10 10 5 1
7:    1 6 15 20 15 6 1
8:   1 7 21 35 35 21 7 1
```

O triângulo de Sierpinski obtém-se do anterior seguindo a mesma representação mas substituindo os elementos ímpares pelo carácter '*' e os pares pelo carácter '.'. Para $n = 8$ o triângulo de Sierpinski é:

```
1:          *
2:         * *
3:        * . *
4:       * * * *
5:      * . . . *
6:     * * . . * *
7:    * . * . * . *
8:   * * * * * * * *
```

Tarefa

Escreva um programa que dado um valor de n ($n \leq 25$) escreva o triângulo de Sierpinski correspondente, alinhado à esquerda e sem espaços.

Input

Uma linha com um inteiro n que corresponde ao tamanho do triângulo de Sierpinski a calcular.

Output

O programa deverá apresentar como resultado o triângulo de Sierpinski alinhado à esquerda e sem espaços, correspondente ao valor de n .

Exemplo: input

8

Exemplo: output

```
*  
**  
*.*  
****  
*...*  
**..**  
*...*  
*****
```

Problema E

Telepreso

Problema

O chefe máximo duma rede de crime organizado está preso numa cadeia de segurança máxima, em isolamento completo, desde há 6 meses. Mesmo assim tem conseguido continuar a gerir a parte da sua organização que ainda não está atrás de grades porque os seus associados fizeram chegar-lhe um telemóvel.



Na realidade apercebemo-nos atempadamente desta intenção mas optámos por não evitar que o telemóvel lhe chegasse às mãos. Esperávamos obter das conversas ao telemóvel informações que nos permitissem fazer outras detenções e colocámo-lo numa cela com microfones ultra sensíveis. No entanto o preso comunica apenas por SMS e para números de telefone no estrangeiro. Para dificultar a situação as mensagens são cifradas automaticamente pelo próprio telemóvel e não as conseguimos decifrar.

Felizmente os microfones instalados na cela são suficientemente sensíveis para distinguirem os tons de cada uma das teclas pressionadas. Para podermos decifrar rapidamente as mensagens que o preso envia, necessitamos dum programa que converta sequências de uma mesma tecla na letra que lhe está associada. Por exemplo, três toques consecutivos da tecla "8" correspondem à letra "V". Mais dois toques na tecla "3", seguidos de dois toques na tecla "6", seguidos de dois toques na tecla "6", seguidos de um toque na tecla "3", seguidos de dois toques na tecla "3", formam a palavra "VENDE".

De notar que as duas últimas letras da palavra "VENDE" ("D" e "E") são conseguidas por sequências da mesma tecla ("3"). Se existir uma certa demora entre o pressionar de duas teclas dos telemóveis assume que a sequência terminou e se iniciou uma nova.

Tarefa

Converter uma sequência de valores numéricos nas letras que lhes estão associadas no teclado dum telemóvel, conforme a seguinte tabela:

Tecla	Repetições			
	1	2	3	4
0	+			
1	-			
2	A	B	C	
3	D	E	F	
4	G	H	I	
5	J	K	L	
6	M	N	O	
7	P	Q	R	S
8	T	U	V	
9	W	X	Y	Z

Algumas teclas podem ser repetidas até três vezes, duas delas podem ser repetidas quatro vezes e duas apenas uma vez. Para facilitar a legibilidade a tecla "1" fica associada ao carácter "_" ("underscore") e não ao carácter espaço. Pode assumir que as teclas não são repetidas mais vezes do que os valores definidos na tabela. Isto é, pode assumir que a tecla "2" nunca é pressionada quatro vezes nem a tecla "1" duas ou mais vezes.

Além dos valores numéricos correspondentes aos dígitos são também usados dois valores numéricos especiais:

- 77 - representa uma espera que permite considerar o fim duma sequência;
- 99 - representa o fim da mensagem.

Input

Uma sequência de linhas, com um inteiro (de 0 a 9 ou 77) por linha, terminada por uma linha com o número 99.

Output

Uma linha com a sequência de letras correspondente aos inteiros lidos.

Exemplo de Input

8
8
8
3
3
6
6
3
77
3
3
1
2
77
1
3
7
7
7
6
6
6
77
4
2
99

Exemplo de Output

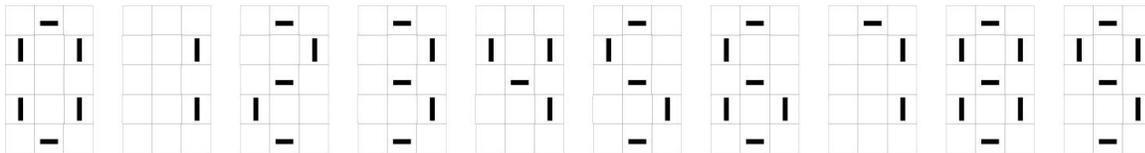
VENDE_A_DROGA

Problema F

Relógio Digital

Problema

Os relógios digitais mais comuns usam os chamados mostradores de sete segmentos, em que cada dígito é construído a partir de três segmentos horizontais e quatro verticais, que podem estar acesos ou apagados. O mapeamento de dígitos em segmentos acesos para os dígitos de 0 a 9 é o seguinte:



Tarefa

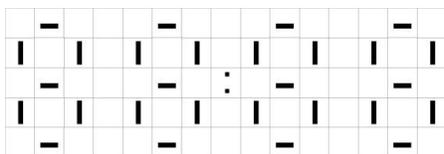
Escrever um programa que, a partir da hora em formato HH:MM (relógio de 24 horas) imprima a correspondente imagem do mostrador de sete segmentos.

Input

A hora em formato HH:MM (relógio de 24 horas).

Output

A imagem do mostrador de sete segmentos, construída a partir dos caracteres “-”, “|” e espaço, com a disposição seguinte:



Repare que entre os dois dígitos das horas, assim como entre os dois dígitos dos minutos, existe uma coluna vertical de espaços; entre as horas e os minutos existe um sinal “:”; todas as linhas de output têm 15 caracteres.

Exemplo: input

12:35

Exemplo: output

```
  - - -  
|  |  | |  
  - : - -  
| |  | |  
  - - -
```

Problema G

Sentar ou não Sentar?

Problema

São de ideias quase fixas os habitantes do reino da Teimosia, preferindo ficar de pé do que sentar-se em cadeiras pelas quais não nutrem especial empatia. Só já muito exaustos se sentam nalguma que, não sendo das suas predilectas, também não rejeitam totalmente. Em sociedade respeitam uma hierarquia estrita, aguardando que os seus superiores se sentem ou insistam em ficar de pé, para avançar então para um lugar da sua preferência, ainda desocupado.



Tarefa

Escrever um programa que dado o número de cadeiras existentes de cada tipo e dadas as preferências dos habitantes, por ordem decrescente de preferência, determina o número dos que ficam de pé e o número de cadeiras que ficam por ocupar. Os habitantes estão exaustos.

Input

A primeira linha tem o número de tipos de cadeiras. A seguir, são dados pares $t \ v$, em que t é um tipo e v o número de cadeiras que existem do tipo t . Se não existirem cadeiras dum dado tipo t , nos dados terá $t = 0$. Note que t e v estão em linhas consecutivas.

Depois da informação sobre as cadeiras, encontra o número de habitantes do reino e , em seguida, as preferências de cada um, estando já indicadas **segundo a hierarquia do reino**. Para cada habitante é dado o número de opções (que pode ser 0) o , em seguida, os tipos de cadeiras em que aceita sentar-se, por ordem decrescente de preferência.

Podem existir até 100 tipos de cadeiras e até 1000 de cada tipo. Nem o número de habitantes do reino nem o número total de cadeiras excede 60000.

Output

Duas linhas: uma com o número de habitantes que ficam de pé e a outra com o número de cadeiras que ficam livres. Terminam por mudança de linha.

Exemplo: input

```
4
1
4
4
0
3
1
2
1
7
2
3
1
4
3
1
2
4
0
1
2
2
3
2
1
1
3
2
4
1
```

Exemplo: output

```
2
1
```