

**CPAS'2007@CIC**

**FINAL**

---

**3 de Maio 2007**  
**Colégio Internato dos Carvalhos**

### Exercício 1

Escreva um programa que sendo dado um número  $X$  inteiro positivo de 3 algarismos ( $100 \leq X \leq 999$ ), permita construir outro número de 4 algarismos, de acordo com a seguinte regra:

- Os 3 primeiros algarismos (contados da esquerda para a direita) são iguais aos do número dado.
- O 4º algarismo é um dígito de controlo calculado da seguinte forma: 1º algarismo + 2º algarismo  $\times$  3 + 3º algarismo  $\times$  5. O dígito de controlo é igual ao resto da divisão dessa soma por 7.

#### Exemplo 1

**Entrada:**

234

**Saída:**

2343

#### Exemplo 2

**Entrada:**

121

**Saída:**

1215

### Exercício 10

Um colecionador de livros raros descobriu recentemente um livro escrito numa língua pouco familiar mas que usa o nosso conjunto de caracteres (considerando apenas letras maiúsculas de A a Z, excluindo os caracteres acentuados e os específicos da língua portuguesa).

O livro possui um pequeno índice, contudo a ordem dos itens do índice é diferente da que se esperaria encontrar caso os caracteres estivessem ordenados da mesma forma que no nosso alfabeto (i.e. ABCDEFGHIJKLMNOPQRSTUVWXYZ).

O colecionador tentou usar o índice para determinar a ordem dos caracteres (i.e. a sequência de junção) do idioma estranho, mas acabou por desistir atendendo a que a tarefa se revelava frustrante e fastidiosa.

A sua tarefa é escrever um programa para completar o trabalho do colecionador. O seu programa deverá receber um conjunto de *strings* ordenadas de acordo com a sequência estabelecida no idioma estranho e determinar essa sequência.

A entrada do programa consiste de uma lista ordenada de *strings*, uma *string* por linha, compostas apenas por caracteres que são letras maiúsculas. Cada *string* contém no máximo 20 caracteres e existem no máximo 100 linhas. O fim da lista está sinalizado por uma linha especial que contém apenas o carácter #. Nem todas as letras são necessariamente usadas, mas uma lista de *strings* dará origem a uma ordenação das letras usadas.

A saída deve ser apenas uma linha com as letras maiúsculas na ordem determinada pela sequência de junção usada para produzir a entrada fornecida.

#### Exemplo 1

**Entrada:**

```
XWY
ZX
ZXY
ZXW
YWX
#
```

**Saída:**

```
XZYW
```

#### Exemplo 1

**Entrada:**

```
FROG
FORG
FGRO
GORF
GFOR
#
```

**Saída:**

```
ROFG
```

### Exercício 2

Alguns números possuem a propriedade de se auto-elogiarem através dos seus dígitos, por esta razão formam uma família de números chamados «narcisistas».

Os narcisistas clássicos são os números que são iguais à soma dos seus dígitos elevados a uma potência igual ao número de dígitos.

Implemente um programa que para um número X inteiro positivo ( $1 \leq X \leq 10000$ ), verifique se é (VERDADEIRO) ou não (FALSO) narcisista clássico.

#### Exemplo 1

$153 = 1^3 + 5^3 + 3^3$

**Entrada:**

153

**Saída:**

VERDADEIRO

#### Exemplo 2

$1634 = 1^4 + 6^4 + 3^4 + 4^4$

**Entrada:**

1634

**Saída:**

VERDADEIRO

#### Exemplo 3

$29 \neq 2^2 + 9^2$

**Entrada:**

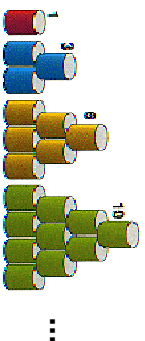
29

**Saída:**

FALSO

### Exercício 3

Implemente um programa que produza uma lista dos  $N$  ( $1 \leq N \leq 100$ ) primeiros números triangulares. Os números triangulares são:



#### Exemplo 1

**Entrada:**

3

**Saída:**

1  
3  
6

#### Exemplo 2

**Entrada:**

10

**Saída:**

1  
3  
6  
10  
15  
21  
28  
36  
45  
55

### Exercício 9

Considerando que uma palavra é uma sequência de caracteres que não contém espaços e um texto constituído por uma única linha com 255 caracteres, no máximo. Implemente um programa que:

- Determine a maior palavra e imprima-a (tal como aparece no texto). Em caso de empate deve considerar a que aparece em primeiro lugar no texto.
- Classifique as palavras quanto ao seu comprimento em 5 classes e determine a frequência absoluta de cada classe (i.e. a contagem das palavras de cada classe). A primeira classe é relativa às palavras com comprimento entre [1-3], a segunda entre [3-6], a terceira entre [6-9], a quarta entre [9-12] e a quinta comprimento maior do que 12.

#### Exemplo 1

**Entrada:**

Era uma vez um gato maltes que tocava piano e falava Frances

**Saída:**

Frances  
6  
5  
1  
0  
0

### Exercício 8

Já jogou o "Campo minado"? Faça um programa que leia uma matriz de caracteres com 5 linhas x 5 colunas. Cada carácter pode ser: \* (asterisco) representa uma bomba na coordenada lida; - (traço) representa um local sem bomba. Crie e imprima uma matriz de inteiros (5 x 5) que contenha para cada posição [i, j] o número de bombas na vizinhança. Cada posição tem no máximo 8 vizinhos (as diagonais contam).

Exemplo de uma matriz lida:

*	-	-	-	-
-	-	-	-	-
*	*	*	*	*
-	-	-	-	*
*	-	-	-	*

Matriz resultado:

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

#### Exemplo 1

Entrada:

```
*-----  
-----  
-----  
*****  
-----*  
*-----*
```

Saída:

```
01000  
34332  
12232  
34353  
01021
```

### Exercício 4

Implemente um programa que leia uma sequência de números naturais no intervalo [1,10000] e verifique se cada um dos números é um número feliz ou um número triste. Exemplos:

- O número 4599 é um número feliz porque  $4^2+5^2+9^2+9^2=203$ ;  $2^2+0^2+3^2=13$ ;  $1^2+3^2=10$ ;  $1^2+0^2=1$ .
- O número 3 é um número triste porque  $3^2=9$  (diferente da unidade).

A sequência termina com o número fora do intervalo [1,10000].

#### Exemplo 1

Entrada:

```
4599  
2007  
12  
0
```

Saída:

```
4599 FELIZ  
2007 TRISTE  
12 TRISTE
```

### Exercício 5

Implemente um programa que leia um universo de 10 células contíguas que constituem a sua população. Cada célula tem um estado que evolui, de geração em geração, de acordo com o estado da própria célula e das suas vizinhas. Considere que cada célula pode ter 3 estados possíveis, 0, 1 ou 2. Cada célula tem um vizinho à esquerda e outro à direita.

As regras de evolução para cada célula são as seguintes:

**O estado é 0** → o estado da célula na próxima geração é 2, se a soma dos vizinhos for igual ou superior a 2, senão permanece a 0.

**O estado é 1** → o próximo estado da célula é 0.

**O estado é 2** → o estado da célula na próxima geração é 1, se algum dos vizinhos for 0, senão permanece a 2.

As células formam um anel, isto é, as células da primeira e da última posição são vizinhas.

#### Exemplo 1

Entrada:

```
0 0 2 1 0 2 1 0 1 1
```

Saída:

```
0 2 1 0 2 1 0 2 0 0
```

### Exercício 6

Considere a string AAAABCCCCDDDD consistindo unicamente de caracteres alfabéticos maiúsculos (excluem-se os acentuados e os específicos da língua portuguesa). Esta string tem o comprimento de 14 caracteres. Como a string contém apenas caracteres alfabéticos, os caracteres duplicados podem ser removidos e substituídos por um factor de duplicação n. Usando esta técnica a string pode ser comprimida e representada por 4AB5C4D (com um comprimento de 7). Escreva um programa que aceite strings no formato comprimido e reconstrua as strings originais.

A entrada será uma série de linhas (no máximo 100) com uma string por linha. Se a string contiver dígitos estes são contadores de compressão. Nunca aparecerão 2 dígitos seguidos. A lista termina com uma string contendo #. As strings nunca terão mais do que 50 caracteres.  
A saída será uma string por linha com a versão descomprimida de cada linha de entrada.

#### Exemplo 1

**Entrada:**

```
3A4B7D
2D7AC8FGD
2ABCDE
#
```

**Saída:**

```
AAAABBBDDDDDDDD
DDAAAAAACFFFFFFFGD
ABCDE
```

### Exercício 7

Prende-se implementar um dicionário de numeração árabe-romana através da utilização de dois vectores, conforme ilustrado. Os elementos de cada um dos vectores com o mesmo índice são sinónimos.

1	4	5	9	10	40	50	90	100	400	500	900	1000
I	IV	V	IX	X	XL	L	XC	C	CD	D	CM	M

Construa um programa que permita converter um número árabe X ( $1 \leq X < 4999$ ) em romano, através da consulta deste dicionário. Ex.: 1947 = 1000 + 900 + 40 + 5 + 1 + 1 = M CM XL V II

#### Exemplo 1

567 = 500 + 50 + 10 + 5 + 1 + 1 = D L X V II

**Entrada:**

567

**Saída:**

DLXVII

#### Exemplo 2

3999 = 1000 + 1000 + 1000 + 900 + 90 + 9 = M M M CM XC IX

**Entrada:**

3999

**Saída:**

MMMCMXCIX