

CPAS'2008@CIC

FINAL

17 de Abril 2008
Colégio Internato dos Carvalhos

Exercício 1

Escreva um programa que converta um número inteiro **X** ($0 \leq X \leq 10000$) para a sua representação em código Morse.

Tabela de conversão dos dígitos de 0 a 9 para código Morse

Dígito	Em Morse
0	-----
1	.-----
2	..-----
3	...-----
4--
5-
6	-----.
7	-----.
8	-----.
9	-----.

Exemplo 1

Entrada:

157

Saída:

.....-.....

Exercício 2

Implemente um programa que leia **N** ($1 \leq N \leq 100$) números na forma **AAAAMMDD** e mostre a data correspondente por extenso. Por exemplo, para o número 20080309 será apresentado 09 DE MARÇO DE 08.

Nota: a saída deve ser apresentada em maiúsculas, sem acentos e sem cedilhhas.

Exemplo 1

Entrada:

3

20070120

19980311

20081231

Saída:

20 DE JANEIRO DE 07

11 DE MARÇO DE 98

31 DE DEZEMBRO DE 08

QUESTÕES SOBRE O FUNCIONAMENTO DO SISTEMA

INPUT/OUTPUT (ENTRADA/SAÍDA)

- ✓ Em todos os problemas o input é feito pelo standard input e o output é feito pelo standard output. Portanto, os vossos programas podem e devem usar as funções "normais" de escrita leitura, como sejam o scanf e printf (em C) ou o read/readln e write/writeln (em Pascal).
- ✓ Os inputs/outputs devem ser exactos, isto é, os dados de entrada e saída devem ser EXACTAMENTE IGUAIS (no conteúdo, na formatação, na sequência, no respeito pelas maiúsculas e minúsculas, no espaçamento, nas mudanças de linha, etc.) aos apresentados no enunciado.
- ✓ O standard input pode ser pensado como uma stream equivalente a qualquer ficheiro. Portanto, quando acaba, tem um end-of-file.
Em C, quando o final de input é atingido, um fgets retorna NULL, um scanf retorna zero argumentos lidos e um feof(stdin) retorna true. Em FP, quando o final de input é atingido, EOF retorna true. Para simular um end-of-file se estiver a escrever via teclado, pressione Ctr+D em Linux e Ctr+Z em Windows.
- ✓ A última linha do output deve incluir mudança de linha (writeln em Pascal ou \n em C).

Importante:

- Em C:
- ✓ Não utilizar fflush(stdin), pois a sua utilização torna todas as submissões erradas.
 - ✓ Deve ter o cuidado de colocar a instrução return 0 na função int main () (o que é interpretado pelo Sistema de Submissão como fim do programa).
 - ✓ São permitidas apenas as bibliotecas: **stdio**, **strings**, **math** e **stdlib**.
- Em FP:
- ✓ Não é permitida a inclusão de qualquer biblioteca externa.

SITUAÇÕES OMITIDAS OU NÃO REGULAMENTADAS SERÃO DECIDIDAS PELO JÚRI DO CONCURSO. EM TODOS OS CASOS, AS EQUIPAS COMPROMETEM-SE A RESPEITAR A DECISÃO SOBERANA DO JÚRI, NÃO HAVENDO DIREITO A RECURSO.

Exercício 3

Implemente um programa que apresente um «quadrado» de dimensão **N** ($1 \leq N \leq 100$), construído de forma a que os elementos das diagonais sejam representados pelo carácter '.' (dois pontos) e os restantes elementos pelo carácter '+' (mais).

Exemplo 1

Entrada:

4

Saída:

```
+++  
+:+  
+:+  
+++
```

Exercício 4

Implemente um programa que leia uma sequência de **N** ($3 \leq N \leq 100$) números inteiros no intervalo [-10000,10000] e indique o número de triplas (subseqüências de 3 números iguais) na seqüência dada.

Exemplo:

Para a seqüência:

4 2 15 15 15 3 7 7 7 2

Deverá ser indicado **3**, pois a seqüência dada contém a tripla formada pelo valor 15 e duas (sobrepostas) formadas pelo valor 7.

Exemplo 1

Entrada:

```
11  
4 2 15 15 15 3 7 7 7 2
```

Saída:

3

Exercício 5

Implemente um programa que calcule a soma de 2 fracções. O programa lê:

- ✓ Linha 1: o sinal da primeira fracção;
 - ✓ Linha 2: o numerador e o denominador da primeira fracção;
 - ✓ Linha 3: o sinal da segunda fracção;
 - ✓ Linha 4: o numerador e denominador da segunda fracção.
- Os numeradores são números inteiros do intervalo [0,10000] e os denominadores são números inteiros do intervalo [1,10000].
- Tenha em consideração os seguintes aspectos na apresentação do resultado:
- ✓ O resultado deve ser apresentado simplificado.
 - ✓ Quando o denominador é 1, deve ser apenas apresentado o numerador da fracção.
 - ✓ Se o resultado for negativo, deve ser apresentado o sinal correspondente (ver exemplo 3).

Exemplo 1: Para as fracções $+ 3/2$ e $+ 2/5$ o resultado é 19/10.

Entrada:

+
3 2
+
2 5

Saída:

19 10

Exemplo 2: Para as fracções $+ 1/3$ e $+ 4/6$ o resultado é 1.

Entrada:

+
1 3
+
4 6

Saída:

1

Exemplo 3: Para as fracções $+ 1/5$ e $- 12/15$ o resultado é $- 3/5$.

Entrada:

+
1 5
-
12 15

Saída:

-
3 5

REGULAMENTO

INÍCIO DE SESSÃO E SOFTWARE

- ✓ Deve efectuar o início de sessão com o utilizador **cpas** e password **cpas123**
- ✓ Todo o software necessário ao desenvolvimento da prova foi previamente instalado, estando os respectivos atalhos disponíveis no ambiente de trabalho.

ADVERTÊNCIAS

- ✓ Não podem consultar quaisquer materiais de apoio, para além dos que forem fornecidos pela organização.
- ✓ Não é permitida a utilização de dispositivos de memória secundária (PenDrives, etc.).
- ✓ Os elementos da equipa podem comunicar apenas entre si, com os cuidados devidos de forma a não perturbar as restantes equipas.
- ✓ Devem desligar o telemóvel ou outros aparelhos de comunicação.
- ✓ Não podem usar a Internet/Intranet para comunicar, consultar informação ou partilhar ficheiros.
- ✓ Só podem abandonar a sala com a autorização do membro da organização presente na sala.
- ✓ Não podem instalar, eliminar ou alterar o software e a configuração dos postos.

No caso do não cumprimento das disposições anteriores a equipa é automaticamente desclassificada.

RECOMENDAÇÕES

- ✓ Por prevenção, devem guardar as resoluções com alguma regularidade no disco local, numa pasta criada para o efeito, com o nome cpas1 para o exercício 1, cpas2 para o exercício 2, etc..

DÚVIDAS

- ✓ Em caso de dúvida, devem dirigir-se ao membro da organização presente na sala.

ATRIBUIÇÃO DA CLASSIFICAÇÃO

- ✓ 100 pontos por cada problema resolvido com sucesso (pressupõe que o programa ultrapassou com sucesso a bateria de testes efectuada automaticamente pelo Sistema de Submissão).
- ✓ Cada problema pode ser submetido 5 vezes, no máximo (esgotadas as 5 tentativas a submissão desse problema fica bloqueada).
- ✓ A classificação final é determinada pela soma dos pontos obtidos nas submissões efectuadas. Em caso de empate, aplicam-se os seguintes critérios:
 1. Tempo gasto pela equipa (em minutos) desde a hora de início do concurso até ao instante em que o último problema é submetido com sucesso.
 2. Após a aplicação do critério anterior, se o empate persistir, o júri decide a classificação em função da qualidade global das soluções apresentadas.

Exercício 10

Implemente um programa que determine e imprima o número de maneiras diferentes que podemos combinar moedas para obter uma dada importância até 1 Euro (incluído). O programa lê a importância a ser obtida, entre [0,100] cêntimos de Euro, e apresenta o número de combinações possíveis.

Moedas existentes:

0,01 Euro	0,02 Euro	0,05 Euro	0,10 Euro
0,20 Euro	0,50 Euro	1 Euro	

Exemplo 1

Entrada:

11

Saída:

12

Exercício 6

Considere-se uma palavra uma sequência contígua de letras (a..z/A..Z) e/ou números. Implemente um programa que leia um texto (apenas com caracteres da língua inglesa) com um número de linhas variável e determine, para cada classe, o número de palavras dessa classe. As classes consideradas são:

CLASSE 1: palavras com número de caracteres no intervalo [1,3].

CLASSE 2: palavras com número de caracteres no intervalo [4,7].

CLASSE 3: palavras com número de caracteres no intervalo [8,11].

CLASSE 4: palavras com número de caracteres superior a 11.

O resultado deve ser apresentada na forma de histograma (gráfico onde cada * representa uma palavra da classe correspondente encontrada no texto).

Exemplo 1

Entrada:

Dia 17 de Abril, no Colegio, sera realizado o
concurso cpas2008@CIC

Saída:

```
CLASSE 1:*****  
CLASSE 2:****  
CLASSE 3:****  
CLASSE 4:
```

Exercício 7

Implemente um programa que, dado um intervalo [A,B] ($2 \leq A \leq B \leq 10000$), determine e apresente qual é o número dentro do intervalo com mais divisores e o seu número de divisores. Se 2 números têm o mesmo número de divisores, deve apresentar o menor número.

Exemplo 1: Sendo A=2 e B=10, o programa apresenta 6 e 4, correspondendo, respectivamente, ao número com mais divisores e o seu número de divisores.

Entrada:

2 6

Saída:

6 4

Exemplo 2: Sendo A=1000 e B=10000, o programa apresenta 7560 e 64, correspondendo, respectivamente, ao número com mais divisores e o seu número de divisores.

Entrada:

1000 10000

Saída:

7560 64

Sugerido por Miguel Oliveira, vencedor do CPAS2005

Exercício 8

Implemente um programa que leia uma lista de nomes completos (máximo 100) e apresente esses nomes de forma abreviada.

Os nomes devem ser abreviados de acordo com as seguintes regras:

- ✓ O primeiro e o último nome são mantidos.
- ✓ Os nomes intermediários são substituídos pela sua letra inicial seguida de ponto, por exemplo, o nome «SAMPALHO» deve ser transformado em «S.».
- ✓ São eliminados os «DE», «DA» e «DOS».

Os nomes são substituídos por letras maiúsculas utilizando apenas os caracteres do alfabeto da língua inglesa. Não existem nomes com mais do que 255 caracteres. A lista de nomes termina com o carácter #.

Nota: nos casos em que, por erro, os nomes foram separados por mais do que um espaço, o programa deve corrigir essa situação, imprimindo apenas um espaço de separação entre os nomes.

Exemplo 1

Entrada:

```
JOAQUIM RIBEIRO CAMPOS
MARIA CONCEICAO ALVES DOS REIS
SUSANA FEITOR
RITA ALEXANDRA MARQUES
#
```

Saída:

```
JOAQUIM R. CAMPOS
MARIA C. A. REIS
SUSANA FEITOR
RITA A. MARQUES
```

Exercício 9

Implemente um programa que com base numa imagem ASCII fornecida como entrada e um factor de ampliação, produza uma nova imagem. As imagens são constituídas pelos caracteres 'X', '/', '\', ' ' e espaços. Cada um destes caracteres pode ser ampliado. Exemplo para o dobro e para o triplo:

```
Normal 2 (dobro) 3 (triplo)
X          \ / \ / \ /
           //  //  //
          \ / \ / \ /
          /  /  /  /
          /  /  /  /
```

Uma ampliação positiva, por exemplo 5, significa que um carácter se transforma num de 5x5. Uma ampliação negativa (redução), por exemplo -3, significa que cada bloco de 3x3 se transforma num carácter.

A entrada do programa consiste numa primeira linha em que são fornecidos 3 inteiros referindo-se, respectivamente, às **C** colunas de largura da imagem, às **L** linhas de altura da imagem e, por último, ao factor **F** de ampliação. Seguem-se as **L** linhas de altura da imagem, cada uma com **C** caracteres. Quer a imagem original, quer a imagem resultante não terão mais do que 500 linhasx500 colunas.

Exemplo 1

Entrada:

```
2 2 2
\X
 X
```

Saída:

```
\ \ \
 \ \ \
  \ \ \
   \ \ \
```

Exemplo 2

Entrada:

```
2 1 2
 X
```

Saída:

```
\ /
 \ /
```

Exemplo 3

Entrada:

```
4 4 -2
 \ \ \ \
  \ \ \ \
   \ \ \ \
    \ \ \ \
```

Saída:

```
\X
 X
```