

#### Departamento de Engenharia Informática e de Sistemas

# Introdução à Programação / Algoritmos e Programação 2010/2011

Data: 13/09/2011 Exame – Época Especial Duração: 2h 30m

### Atenção:

- ✓ Para cada um dos problemas propostos no exame deve apresentar:
  - a <u>análise do problema</u>, especificando os dados de entrada, os resultados pretendidos e as acções a efectuar
  - o respectivo algoritmo em pseudocódigo
  - a implementação do programa em linguagem C.
- ✓ As funções desenvolvidas devem ser portáveis.
- ✓ A prova é sem consulta.
- ✓ Resolva uma pergunta por folha de prova.
- **1.** (5.0 valores) Desenvolva um programa que determine o menor valor positivo de uma sequência de valores inteiros lidos a partir do teclado. A leitura da sequência termina quando for introduzido o valor zero (0).

O programa deve também informar a posição em que o menor valor positivo ocorreu na sequência lida. Se esse valor existir mais do que uma vez na sequência de valores inteiros lidos, deve assinalar a posição da última ocorrência.

#### Exemplo de execução do programa:

```
Diga um número inteiro: 6
Diga um número inteiro: -12
Diga um número inteiro: 10
Diga um número inteiro: 120
Diga um número inteiro: 2
Diga um número inteiro: -18
Diga um número inteiro: -8
Diga um número inteiro: 14
Diga um número inteiro: 2
...
Leitura dos restantes valores inteiros da sequência
...
Diga um número inteiro: 0
O menor valor positivo lido é o 2 e surgiu na posição 9
```

*Nota*: Esta questão deve ser resolvida <u>sem utilizar "arrays"</u>. Resoluções que façam uso destas estruturas de dados, serão cotadas com zero valores.

# 2.

a) (3,5 valores) Desenvolva uma função que desenhe um trapézio (ou quadrado) de caracteres. A função recebe quatro parâmetros: o primeiro parâmetro corresponde ao carácter a usar no desenho do trapézio, car; o segundo parâmetro representa o número de caracteres que compõem o topo do trapézio, topo; o terceiro parâmetro representa o número de caracteres que compõem a base, base; e o quarto parâmetro representa o número de caracteres em branco que devem ser desenhados antes da base do trapézio de modo a ele ficar posicionado no sítio desejado.

No desenho do trapézio (**topo** diferente de **base**), no sentido topo para base, cada linha é composta pelo número de caracteres da linha anterior + 2. Os valores do **topo** e da **base** são inteiros ímpares.



#### Departamento de Engenharia Informática e de Sistemas

De seguida são apresentados dois exemplos de execução da função para melhor compreensão do enunciado ('·' representa o espaço em branco e '¶' a mudança de linha).

# Resultado de execução da função (car=#, topo=3, base=11, extra=2):

```
P### - - - - - . . .
P##### - - - - . .
P##### - - - . .
P##### - - - .
P###### - - .
P###### - - . .
P###### - . .
```

# Resultado de execução da função (car=T, topo=3, base=3, extra=6):

```
TTT¶
.....TTT¶
.....TTT¶
```

b) (5,5 valores) Elabore um programa que, recorrendo à função da alínea a), permita construir uma árvore de Natal com um determinado número de andares. Para esse efeito, o programa deverá ler o número de andares da árvore de Natal, e chamar a função tantas vezes quantas o número de andares mais uma (para desenhar o tronco). Assuma que o andar de cima é sempre especificado pelos parâmetros car = #, topo = 3, base = 11, sendo o valor do parâmetro extra calculado de acordo com o número de andares (zero para uma árvore com apenas um andar), por forma a que a árvore fique alinhada do lado esquerdo do monitor.

#### Exemplo de execução do programa:

```
Introduza o número de andares da árvore de Natal: 3
....###
....#####
.....#######
. . . . . #########
. . . . ###########
 ....#######
   .###########
. . . . ############
  ###############
  ################
####################
.......TTT
......TTT
....TTT
Construí uma árvore de Natal com 3 andares e 1 tronco!!!
```

**3.** (5.5 valores) Construa um programa que solicite ao utilizador a introdução de uma string com tamanho máximo de 100 caracteres e de uma palavra com um máximo de 20 caracteres.

O programa deve permitir substituir a primeira palavra na *string*, cujo número de caracteres coincida com o número de caracteres da palavra introduzida pelo utilizador (excepto '\0'). Caso não exista nenhuma palavra que verifique este critério, o programa deverá produzir a seguinte mensagem: "A frase não foi alterada".

#### Exemplo:

String: vamos considerar ola um exemplo!

Palavra: sou

String alterada: vamos considerar sou um exemplo!