

## 21010 - Arquitetura de Computadores

### Enunciado

Considere um vetor de inteiros, o qual se pretende ordenar. Antes de ser chamada uma rotina para processar o vetor é colocado no stack o endereço do primeiro elemento, seguido do número de elementos do vetor.

a) [1] Desenvolva uma sub-rotina no Assembly do P3 que tendo como entrada o vetor indicado, coloque no registo R1 a indicação de se o vetor está ordenado (valor 0001h) ou não (valor 0002h).

b) [1] Desenvolva uma sub-rotina no Assembly do P3 que receba como entrada o vetor indicado e retorne o vetor com os elementos ordenados por ordem crescente usando o algoritmo “bubble sort”.

O algoritmo “bubble sort” faz tantas passagens pelo vetor quantos os elementos do vetor. Cada passagem começa no início do vetor e compara cada par de elementos adjacente e caso estejam por ordem decrescente, troca um elemento com outro. A primeira passagem garante que o maior elemento fica no final do vetor, com um efeito parecido com uma “bolha”, daí o nome do algoritmo. Cada passagem garante que mais um elemento fica por ordem.

Mais informação consultar [https://pt.wikipedia.org/wiki/Bubble\\_sort](https://pt.wikipedia.org/wiki/Bubble_sort) ou [https://en.wikipedia.org/wiki/Bubble\\_sort](https://en.wikipedia.org/wiki/Bubble_sort)

c) [1] Desenvolva uma sub-rotina do Assembly do P3 que receba três vetores dos quais dois estão ordenados por ordem crescente e o terceiro contém o espaço para construir um vetor ordenado por ordem crescente contendo todos os elementos dos outros dois vetores de entrada.

d) [1] Desenvolva uma sub-rotina no Assembly do P3 que recebe como entrada o vetor indicado e retorne o vetor com os elementos ordenados por ordem crescente usando o algoritmo “merge sort”.

O algoritmo “merge sort” é um algoritmo recursivo que em cada passo divide o vetor inicial em dois de metade da dimensão (ou sensivelmente metade) e ordena cada um dos vetores menores, juntando (ou fazendo “merge”) após a ordenação de ambos os vetores. Os vectores vão sendo divididos até se chegar à situação de vectores de um só elemento, que estão necessariamente ordenados. Mais informação consultar [https://pt.wikipedia.org/wiki/Merge\\_sort](https://pt.wikipedia.org/wiki/Merge_sort) ou [https://en.wikipedia.org/wiki/Merge\\_sort](https://en.wikipedia.org/wiki/Merge_sort)

Desenvolva as suas sub-rotinas no programa base seguinte:

```
; Ficheiro de Assembly do P3
;
; Programa para desenvolvimento do e-fólio B.
;
; O vetor inicia-se na posição de memória 8000h,
; sendo o número de elementos N_ELEM.
;
; No registo R1 retorna-se 1 se o vetor estiver ordenado
; por ordem ascendente e 2 em caso contrário.

;Constantes

ORD EQU 0001h ;vetor ordenado
NORD EQU 0002h ;vetor não ordenado
N_ELEM EQU 7 ;número de elementos do vetor

ORIG 8000h
VETOR STR 2,3,4,5,6,1,8
ORIG 0000h

;Inicialização do stack pointer (SP)

Inicio: MOV R1, FDFH
        MOV SP, R1

;Programa

        PUSH VETOR
        PUSH N_ELEM
        CALL ALINEA_A

        PUSH VETOR
        PUSH N_ELEM
        CALL ALINEA_B
```

```
PUSH VETOR
PUSH 5
PUSH 8005h
PUSH 2
PUSH 8010h
PUSH N_ELEM
CALL ALINEA_C

PUSH VETOR
PUSH N_ELEM
PUSH 8020h
CALL ALINEA_D

FIM: JMP FIM

;Subrotina ALINEA_A

ALINEA_A:
; Escreva aqui a sua sub-rotina
AA_FIM: RETN 2

;Subrotina ALINEA_B

ALINEA_B:
; Escreva aqui a sua sub-rotina
AB_FIM: RETN 2

;Subrotina ALINEA_C

ALINEA_C:
; Escreva aqui a sua sub-rotina
AC_FIM: RETN 6

;Subrotina ALINEA_D

ALINEA_D:
; Escreva aqui a sua sub-rotina
AD_FIM: RETN 3
```

**BOM TRABALHO!**

## Avaliação

### Cotação:

A cotação encontra-se junto de cada uma das alíneas, entre [].

### CrITÉRIOS de Correção:

Funcionalidade: 50%

Simplicidade e Modularidade: 10%

Eficiência (serão contabilizados o número de instruções e ciclos de relógio): 10%

Apresentação do código (indentação e comentários): 20%

Relatório (Legibilidade e Justificação dos Resultados e das Opções): 10%

### Descontos:

Trabalhos entregues que não estejam em conformidade com as regras de entrega do e-fólio B: até 10%

Código sem comentários, ou apenas com comentários a reflectir o significado da instrução (exemplo MOV R1,R2 ;mover o conteúdo de R2 para R1) : até 50%

Deteção de fraude (total ou parcial): 100%

Trabalhos entregues após a data limite (o recurso de entrega estará aberto até ao início da manhã do dia seguinte, no entanto entregas após a data/hora indicada, têm desconto na nota) : 10%

## Regras para entrega do e-fólio B:

### Forma de entrega:

Um ficheiro zipado com o nome correspondente ao número de aluno.

O ficheiro zipado deverá por sua vez conter um ficheiro de Assembly do P3 por alínea, cujo nome será o número de aluno mais a letra da alínea ex: 999999a.as.

Deverá ainda ser entregue um pequeno relatório em formato pdf de até 5 páginas A4, com todos os cálculos e todas as opções tomadas na construção dos programas.

Regras relativas à escrita dos programas de Assembly:

Todos os programas devem acabar com a seguinte instrução:

FIM:           JMP FIM                   ;Fim do programa

Não são aceites entregas fora da plataforma Moodle.