

e-Fólio B

21010 - Arquitectura de Computadores, UAb, 2013/2014

Enunciado

Pretende-se que desenvolva um programa em Assembly do P3, que processe um vector de inteiros de acordo com a funcionalidade pedida em cada uma das 3 alíneas. O vector é enviado para as funções de cada alínea, através do endereço da primeira posição de memória, que contém nessa posição o número de elementos no vector. É dado um programa base em anexo, com a entrada de dados, e três chamadas a funções, uma para cada alínea. Os comentários podem ser actualizados, mas o código inicial deve ser considerado fixo, pelo que, se existir necessidade de alterá-lo, deve ser dada uma justificação dessa alteração no relatório.

Alínea A – SomaVetor (1,5 valores)

Esta alínea recebe um argumento, com o vector. Deve somar todos os elementos e colocar o resultado em R1. A primeira posição do vector não deve ser somada, já que indica quantos elementos tem o vector. Caso a soma caiba numa posição de memória do P3, 16 bits, retornar em R1 o resultado, caso contrário colocar em R1 o valor FFFFh.

Alínea B – HistogramaVetor (1,5 valores)

Pretende-se nesta alínea, que sejam processados os dados do primeiro argumento enviado para a função, e colocado o resultado no vector de 4 elementos, fornecido no segundo argumento. Deve ser feito um histograma com 4 intervalos equidistantes, e em cada intervalo (posição do vector), deverá estar o número de elementos que estiverem no intervalo. Os intervalos devem estar entre o valor mínimo e máximo dos dados, como se exemplifica de seguida:

- Dados: 20; 24; 40; 35; 36; 37.
- Valor mínimo: 20
- Valor máximo: 40
- Intervalos: 20-25; 25-30; 30-35; 35-40.
- Histograma: 2; 0; 1; 3.

Notar que o valor 35, tanto poderia ser colocado no 3º como no 4º intervalo. Não poderia é ser colocado em ambos os intervalos, e também não poderia deixar de ser colocado num dos intervalos.

Alínea C – QuickSort (1 valor)

Nesta alínea, deve ordenar o vector pelo método quicksort (<http://pt.wikipedia.org/wiki/Quicksort>), podendo fazer alterações para otimizar o algoritmo. Será publicado uma lista das execuções apenas com esta alínea, com menor número de ciclos, e outra lista com o menor número de instruções executadas, necessárias para ordenar ambos os vectores fornecidos. Durante o e-fólio não devem ser fornecidos resultados, de modo a não

dar referências aos restantes colegas. O vencedor ganhará 1 valor extra, e quem obter uma melhor performance que a resolução de referência, ganhará 0,5 valores extra. Os valores extra são somados na avaliação contínua, até ao máximo de 8 valores.

Código Base

Código enviado também em ficheiro anexo:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Efolio B 2013/14
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Dado um vector com inteiros, processar
; (primeiro valor tem o número de inteiros)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Alínea A: retorna a soma dos números
; Alínea B: retorna histograma
; (4 barras, número de elementos em cada intervalo)
; Alínea C: ordena com o método quicksort
; (http://pt.wikipedia.org/wiki/Quicksort)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

ORIG 8000h

; espaço onde colocar os resultados das alíneas
; Alínea A
Soma WORD 0
; Alínea B
Histograma TAB 4
; Alínea C: alterar o próprio vector, colocando-o ordenado

; Dados a processar
Vector STR 100, 2284, 9017, 3891, 4459, 2475, 2617, 5347, 8364...
Vector2 STR 40, 2, 6, 4, 9, 7, 4, 9, 5, 4, 9, 6, 9, 8, 9, 8...

ORIG 0000h

; inicialização do stack
MOV R1, fe00h
MOV SP, R1

; chamar as alíneas por ordem
PUSH Vector
CALL SomaVetor ; retornar resultado em R1
MOV M[Soma], R1

PUSH Vector
PUSH Histograma ; local onde retornar o resultado
CALL HistogramaVetor

PUSH Vector
CALL QuickSort

PUSH Vector2
CALL QuickSort

Fim: BR Fim

; Atenção: utilizar nas alíneas os parâmetros passados pelo stack
; não utilizar: Soma; Histograma; Vector; Vector2

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Alínea A: retorna a soma dos números
; retornar o valor em R1
SomaVetor: MOV R1, R0
RETN 1 ; remover o parâmetro passado pelo stack

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Alínea B: retorna histograma
; (4 barras, número de elementos em cada intervalo)
; retorna o histograma no segundo argumento passado
HistogramaVetor: MOV R1, R0
RETN 2 ; remover os dois parâmetros passados pelo stack

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Alínea C: ordena com o método quicksort
; altera o vetor directamente
QuickSort: MOV R1, R0
RETN 1 ; remover o parâmetro passado pelo stack
```

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 que não estejam em conformidade com as regras de entrega: 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%
- Detecção de fraude (total ou parcial): 100%
- Entregas tardias (até 24h): 10%

Regras de entrega:

- Um ficheiro zipado com o número do estudante;
- O ficheiro zipado deverá conter um ficheiro de Assembly do P3, feito com base no ficheiro base.as. O nome do ficheiro deverá ser o número do estudante.
- Deverá entregar um pequeno relatório em formato pdf até 5 páginas A4, com todos os cálculos e todas as opções tomadas na construção do programa;
- Não são aceites entregas fora da plataforma Moodle.