

Atenção: Este exame tem 2 partes, num total de 200 pontos.

A **parte 1** é composta por 10 questões de escolha múltipla a serem respondidas no **MOODLE**.

A **parte 2** é composta pelas 2 questões de programação apresentadas a seguir. As sub-rotinas pedidas devem ser submetidas no **SIGEX**.

Durante a prova apenas poderá ser utilizada a folha de consulta de instruções assembly disponível no Moodle. Para realizar os exercícios de programação utilize o seu editor de texto preferido.

Parte 2 — Questões de programação

- [50] 1. Dados dois vetores de números inteiros ($v[]$ e $w[]$) com o mesmo comprimento (n), pretende-se calcular quantos pares de valores (um de cada vetor) com o mesmo índice estão próximos. Os dois valores de um par estão próximos se o valor absoluto da sua diferença for inferior ou igual a uma dada distância ($dist$).

Em C, a sub-rotina `nperto` que realiza esta tarefa tem a seguinte declaração:

```
unsigned int nperto(int v[], int w[], unsigned int n, unsigned int dist);
```

Exemplos:

$v[]$	$w[]$	n	$dist$	resultado
{ 6, 0, -3 }	{ 9, 2, -4 }	3	2	2
{ -2, 4, 8, 9 }	{ 6, -5, 2, 12 }	4	5	1

Escrever a sub-rotina `perto` em linguagem *assembly* e submeter o resultado via **SIGEX** conforme indicado abaixo.

Observações importantes:

- O início da sua sub-rotina deverá ser:

```
.text
.global nperto
.type nperto, %function
nperto:
// restante código
```

- Para testar a sua solução durante o exame aceda à plataforma de testes (<https://aoco.fe.up.pt/>) e:
 - Selecione o exercício “EXAME 1”.
 - Coloque a sua sub-rotina `nperto` num ficheiro chamado **nperto.s** e depois crie um ficheiro **.zip** contendo apenas o ficheiro da sub-rotina. (O nome dado ao ficheiro **.zip** não é relevante.)
 - Submeta o ficheiro para teste.
- Para submissão da resposta no **SIGEX**, a sub-rotina `nperto` deve ser colocada num ficheiro designado `nperto_upXXXXXXXX_1MIEIC0x.s`, em que “XXXXXXXXXX” é o código do estudante e “x” é o número da turma.

Resposta:

```
.text
.global nperto
.type nperto, %function

// Entrada
// X0: endereço-base de v[]
// X1: endereço-base de W[]
// W2: n.º de elementos
// W3: distância máxima
// Saída
// W0: número de pares de valores próximos

nperto:
    mov     W4, 0      // inicializar contador de ocorrências
L1:
    cbz     W2, LFim
    ldr     W5, [X0], 4    // ler 1.º valor e ajustar apontador
    ldr     W6, [X1], 4    // ler 2.º valor e ajustar apontador
    subs    W5, W5, W6    // calcular a diferença e alterar flags
    cneg    W5, W5, MI    // valor absoluto
    cmp     W5, W3        // sem sinal
    csinc   W4, W4, W4, HI
    sub     W2, W2, 1
    b       L1
LFim:
    mov     W0, W4
    ret
```

[50] 2. Considerar a função $f(x)$, $x \in \mathbb{R}$, definida por:

$$f(x) = \begin{cases} y(x) & x < 0 \\ \sqrt{5 + x^2} & x \geq 0 \end{cases}$$

Implementar a sub-rotina **sumf** que calcula o valor da função $f(x)$ para cada valor da sequência **seqA**, devolvendo o somatório dos valores calculados.

O cálculo do ramo correspondente a $x < 0$, (i.e., função $y(x)$) é realizado pela sub-rotina **funky**, a ser invocada a partir de **sumf**. A sub-rotina **funky** recebe um valor em D0 e devolve em D0 o resultado.

A declaração da sub-rotina para uso num programa escrito em C é:

```
double sumf(double *seqA, int tamA);
```

Exemplos:

1. se $\text{seqA} = \{2.0, 3.0, 3.0, 2.0\}$ e $\text{tamA} = 4$, o resultado será 13.483315;
2. se $\text{seqA} = \{-2.0, 3.0, 3.0, -2.0\}$ e $\text{tamA} = 4$, o resultado será -8.516685.

Observações importantes:

- O início da sub-rotina deverá ser:

```
.text
.global sumf
.type sumf, %function
sumf:
// restante código
```

- Para testar a sua solução durante o exame aceda à plataforma de testes (<https://aoco.fe.up.pt/>) e:
 1. Selecione o exercício “EXAME 2”.
 2. Coloque a sua sub-rotina **sumf** num ficheiro chamado **sumf.s** e crie um ficheiro **.zip** contendo apenas o ficheiro da sub-rotina (o nome dado ao ficheiro **.zip** não é relevante).
 3. Submeta o ficheiro para teste.
- Para submissão da resposta no **SIGEX**; a sub-rotina **sumf** deve ser colocada num ficheiro designado **sumf_XXXXXXXX_1MIEIC0x.s**, em que “XXXXXXXX” é o código do estudante e “x” é o número da turma.

Resposta:

sumf:

```
    stp    x29, x30, [sp, -64]!
    mov    x19, sp

    mov    W2, 0
    scvtf  D1, W2 // somatório
    scvtf  D2, W2 // zero para comparacao
    mov    W3, 5
    scvtf  D3, W3 // valor 5 para calculo
```

ciclo:

```
    cbz    W1, fim
    ldr     D4, [X0], 8
    sub     w1, w1, 1
    fcmp    D4, D2
    b.lt    callFuncY
    fmul    D4, D4, D4
    fadd    D4, D4, D3
    fsqrt   D4, D4
    fadd    D1, D1, D4
    b       ciclo
```

callFuncY:

```
    fmov    D0, D4
    stp     X0, X1, [sp, 16] // guarda conteúdo dos registo na pilha
    stp     D1, D2, [sp, 32]
    str     D3, [sp, 48]
    bl      funcy
    ldp     X0, X1, [sp, 16] // recuperar valores dos registos
    ldp     D1, D2, [sp, 32]
    ldr     D3, [sp, 48]
    fadd    D1, D1, D0
    b       ciclo
```

fim:

```
    fmov    D0, D1
    ldp     x29, x30, [sp], 64
    ret
```

Fim.