

Atenção: Este teste tem 13 questões em 5 páginas, num total de 200 pontos.

Parte I – Questões de Escolha Múltipla

Cada questão tem uma resposta certa. Respostas erradas não descontam.

As respostas às questões de escolha múltipla devem ser assinaladas com X na grelha da página 4.

Apenas as respostas indicadas na grelha são consideradas para efeitos de avaliação.

- [10] 1. Considere que $SP=0x0805034A0$ e que após execução das instruções seguintes o estado da pilha é o indicado na tabela.
- | | Endereço (hex.) | Conteúdo (hex.) |
|------------------------------|-----------------|-----------------|
| SMADDL X1, W1, W1, X1 | 0805034A8 | 0FF |
| STR X1, [SP, #-16]! | 0805034A0 | 001 |
| | 080503498 | 000 |
| | 080503490 | 00C |

Nestas circunstâncias pode afirmar-se que o valor inicial de X1 é:

- A. 1 B. -1 C. 0 **D. 3**

- [10] 2. Considere que $W3=0x8ABC0DEF$ antes de executar o código seguinte. Indique o valor final de W3.

```
str    W3, [X0]
ldrsb  W3, [X0]
```

- A. 0x8ABC0DEF **B. 0xFFFFFEF** C. 0x000000EF D. 00000DEF

- [10] 3. Considere o seguinte programa composto por código C e *assembly* AArch64.

```
extern unsigned long int SUBR_T2(char *p);

int main(void) {
    char s[] = "30 de Abril";
    unsigned long int res;

    res = SUBR_T2(s);
    printf("0x%x\n", res); //Imprime em hexadecimal
    return EXIT_SUCCESS;}

    .text
    .global SUBR_T2
    .type SUBR_T2, "function"

SUBR_T2: LDRB W1, [X0]
        CBZ W1, FIM
        ADD X0, X0, #1
        B SUBR_T2
FIM:    RET
```

Após execução é escrito no ecrã 0x0000000FEFFFD3.

Qual o endereço de memória ocupado pelo primeiro elemento ('3') da cadeia de caracteres s?

- A. 0x0000000FEFFFD2 **B. 0x0000000FEFFFC8**
C. 0x0000000FEFFFC7 D. 0x0000000FEFFFB8

- [10] 4. Qual das seguintes instruções dá erro de compilação?

- A. SUB X2,X10,X2,ASR #2 B. ANDS W1,W2,W3 C. SMULH X0,X0,X0 **D. LDR W1,[W0,#4]**

- [10] 5. Suponha que $x0=0x10000000$ e que a tabela representa o conteúdo da memória após executar a instrução `str w9,[x0]`. Qual o conteúdo inicial de `w9` ?

Endereço (hex)	Conteúdo (hex)	
10000003	0F	A. 0x0F31600C
10000002	31	B. 0xC00613F0
10000001	60	C. 0xF01306C0
10000000	0C	D. 0x0C60310F

- [10] 6. Qual das seguintes variantes provoca um erro de compilação?

A. `LDR X9,[X0]` **B. `LDRB X9,[X0]`** C. `LDRSW X9,[X0]` D. `LDRSB X9,[X0]`

- [10] 7. Considere a declaração `extern int cp(int x);` e a respetiva rotina em *assembly* AArch64:

<pre>cp: mov w2,#1 eor w3,w3,w3 lpp: cbz w0,stop and w4,w0,w2 add w3,w3,w4</pre>	<pre> lsr w0,w0,#1 b lpp stop: and w0,w3,w2 ret</pre>
---	---

Assumindo que a função `cp()` é aplicada a cada elemento da sequência (1,3,5,7) quais os resultados obtidos ?

A. (1,1,1,1) B. (0,0,0,0) C. (1,0,1,0) **D. (1,0,0,1)**

- [10] 8. A instrução `ROR X5, X5, #32` permuta as duas *words* de `X5`. Indique o fragmento de código que não realiza esta operação.

<p>A. <code>UBFX X6, X5, #32, #32</code> <code>BFI X5, X5, #32, #32</code> <code>ORR X5, X5, X6</code></p> <p>C. <code>MOV X6, X5</code> <code>BFI X5, X6, #32, #32</code> <code>UBFX X5, X6, #32, #32</code></p>	<p>B. <code>REV X5, X5</code> <code>REV32 X5, X5</code></p> <p>D. <code>MOV W6, W5</code> <code>LSR X5, X5, #32</code> <code>ADD X5, X5, X6, LSL #32</code></p>
---	--

- [10] 9. Considere a declaração `extern int ism(int x, int y);` e respetiva rotina em *assembly* AArch64.

<pre>ism: sdiv w2,w0,w1 msub w0,w1,w2,w0 cmp w0,#0</pre>	<pre> cset w0,eq ret</pre>
---	--

Qual o valor de $m = \text{ism}(34,7)$?

A. 1 B. 4 **C. 0** D. 6

- [10] 10. Considere a declaração `extern int vsum(int *a, int n);` e respetiva rotina em *assembly* AArch64 que pretende calcular a soma dos elementos de um vetor.

<pre>vsum: eor x10,x10,x10 nxt: cbz x1,stop ldrsw x9,[x0],#4 add x10,x10,x9</pre>	<pre> sub x1,x1,#1 b nxt stop: mov x0,x10 ret</pre>
--	--

Qual das afirmações é verdadeira?

A. **A rotina está correta.** B. A rotina está errada pois retorna $n \times v[1]$.
 C. A rotina está errada pois retorna $n \times v[0]$. D. A rotina está errada pois retorna $(n - 1) \times v[1]$.

11. Considere o seguinte programa composto por dois ficheiros.

Ficheiro em linguagem C (main.c):

```
extern int SUBROT1(int *a, int d);
int main(void)
{
    int n = 0, tam = 5;
    int seq[] = {1, 3, 6, 1, 9};
    n = SUBROT1(seq, tam);
    printf("%d\n", n);
    return EXIT_SUCCESS;
}
```

Ficheiro em linguagem *assembly*:

<pre> SUBROT1: STP X29, X30, [SP, #-16]! //<1> MOV X29, SP MOV W10, 0 LDR W12, [X0] ADD X0, X0, #4 SUB X1, X1, #1 MOV W11, #1 CICLO: CBZ X1, FIM LDR W13, [X0] CMP W13, W12 B.LE FSC //<2> ADD W11, W11, #1 B PROX </pre>	<pre> FSC: CMP W10, W11 B.GE PROX MOV W10, W11 MOV W11, #1 PROX: MOV W12, W13 ADD X0, X0, #4 SUB X1, X1, #1 B CICLO FIM: MOV W0, W10 CMP W10, W11 B.GE TERMINAR MOV W0, W11 TERMINAR: LDP X29, X30, [SP], #16 RET </pre>
---	--

- [10] (a) Quanta informação, em número de palavras (*words*), é lida de memória ?
A. 9 B. 5 C. 6 D. 7
- [10] (b) Assumindo que o valor inicial do registo SP é 0xE0, indique em que endereço da pilha se encontra armazenado o registo X30 após a execução da instrução assinalada com <1>.
A. 0xD8 B. 0xD0 C. 0xE0 D. 0xE8
- [10] (c) Para os valores de invocação indicados em main.c, quantas vezes é tomado o salto assinalado com <2>?
A. 2 **B. 1** C. 3 D. 4
- [10] (d) Para os valores indicados no programa, qual é o resultado apresentado no ecrã?
A. 5 B. 2 **C. 3** D. 9

Nome: _____ Nº de estudante: _____

Parte I – Grelha de respostas

Opção	Questão													
	1	2	3	4	5	6	7	8	9	10	11a	11b	11c	11d
A					×			×		×	×	×		
B		×	×			×							×	
C								×	×					×
D	×			×			×							

Pontos: _____ /140

Parte II – Exercícios de desenvolvimento

- [30] 12. A distância de Hamming $\mathcal{H}(a, b)$ entre dois números a e b é dada pelo número de posições em que as respectivas representações binárias são diferentes. Por exemplo, $\mathcal{H}(7, 13) = 2$, conforme ilustrado a seguir:

7	0	0	0	0	0	1	1	1
13	0	0	0	0	1	1	0	1

Escrever a sub-rotina DHAMMING, que determina a distância de Hamming entre dois valores de 64 bits.

Os argumentos da sub-rotina são dois números de 64 bits (tipo DWORD); o resultado é um número inteiro do tipo WORD.

```

DHAMMING:
    eor    x1, x1, x0
    // contar número de bits a 1
    mov    w0, 0
L1:
    cbz    x1, Lf
    ands   x2, x1, 1
    b.eq   L2
    add    w0, w0, 1
L2:
    lsr    x1, x1, 1
    b      L1
Lf:
    ret

```

Nome: _____ Nº de estudante: _____

- [30] 13. Escrever a sub-rotina L3A que recebe uma sequência de N números inteiros (com sinal) do tipo WORD e retorna o maior valor L resultante da adição de três elementos sucessivos da sequência.

A sub-rotina recebe, por ordem, o endereço-base da sequência e o seu número de elementos N (um valor positivo do tipo WORD, $N > 2$), retornando também um valor do tipo WORD.

Exemplo: A sequência

0	8	4	12	2	10	7	15	3
---	---	---	----	---	----	---	----	---

 ($N=9$) resulta em $L = 32$ (ver figura abaixo).

0	8	4	12	2	10	7	15	3
		12	24	18	24	19	32	25

```
// Uma das soluções possíveis
//IN: X0 - endereço base
// W1 - Número de elementos (> 2)
//OUT: W0 - Maior soma de três elementos consecutivos
L3A:
    ldr    w2, [x0], 4
    ldr    w3, [x0], 4
    ldr    w4, [x0], 4
    add    w5, w2, w3
    add    w5, w5, w4
    mov    w6, w5          // máximo corrente
    sub    w1, w1, 3       // n. adições ainda a efetuar
LX:
    cbz    w1, L3A_fim
    mov    w2, w3
    mov    w3, w4
    ldr    w4, [x0], 4
    add    w5, w2, w3
    add    w5, w5, w4
    cmp    w5, w6
    csel    w6, w5, w6, GT
    sub    w1, w1, 1
    b      LX
L3A_fim:
    mov    w0, w6
    ret
```