

NOME:

Nº:

Notas:

1. Para cada uma das **5 questões de resposta satisfatória obrigatória**, numeradas de **1 a 5**, são-lhe oferecidas pelo menos 2 hipóteses para responder e/ou comentar; para cada um destas deverá optar por responder a **apenas uma** delas.
2. Para cada uma das hipóteses que optar, deverá apresentar a **justificação da solução**, incluindo o raciocínio ou os cálculos que efetuar.
3. **Não são permitidas:** (i) máquinas de calcular e (ii) notas auxiliares de memória.
4. Correção de cada questão: não-satisfaz (0), satisfaz com erros (0.8), certa com falhas (1.0) e completamente certa (1.2).

1.

- a) Pretende-se armazenar numa única sequência de bits os números dos lugares de uma sala de cinema, organizados em filas e colunas. Sabendo que esta sala tem 74 filas de lugares, organizados em 18 colunas, sugira uma forma compacta de codificar esta informação indicando quantos bits são necessários e exemplificando para a fila 53, coluna 2.
- b) Pretende-se desenvolver uma codificação em binário para os caracteres de uma linguagem que apenas utiliza vogais e o espaço. Sugira uma codificação que minimize o número de bits utilizado por carácter e apresente a forma como se representa a cadeia "AU EI".

2.

- a) Considere o seguinte fragmento de código em C:

```
int i=10;
do {
    // bloco de código
    i--;
} while (i>5);
```

Complete o mesmo fragmento compilado para *assembly* introduzindo comentários (note que deve indicar qual a condição a testar pela instrução de salto condicional)

```
.L1:
    # bloco de código
    subl ____, %ebx
    cmpl $5, ____
    j____ .L1
```

- b) Represente a estrutura genérica de código C a que corresponde o seguinte fragmento de código *assembly*:

```
cmpl $10, %ebx
jle .L2
imull $2, %ebx
jmp .L3
.L2:
imull $3, %ebx
.L3:
```

Nota: Para as seguintes 3 questões escolha apenas uma afirmação, indique se **Verdadeira** ou **Falsa**, e justifique a sua resposta.

3. Considere o processador IA-16, semelhante ao do IBM PC original (inteiros: 16-bits em complemento para 2).

- a) Se no registo `%ax` estiver o inteiro `0xff`, então ele contém o valor -1 em decimal.
- b) Se usasse um programa em C para calcular a diferença entre os inteiros `x1` e `x2`, contendo respetivamente a circunferência em km da Terra no Equador (40,075) e passando pelos Pólos Norte e Sul (40,008), iria apresentar o resultado em hexadecimal `0x43`, correspondente a 67 (km).

4. Considere uma versão reduzida da norma IEEE 754 com 16 bits (6 bits para o expoente em excesso de 31, 9 para a mantissa e 1 para o sinal; não esquecer os casos de exceção). O valor decimal de um n^o normalizado representado com este formato vem dado por $V = (-1)^S * 1.F * 2^{(Exp-31)}$

- a) Se tiver em registo `0x5c00` como sendo a dimensão da cache L1 de um dado processador (neste tipo de representação), ele representaria a dimensão 32k
- b) Esta versão reduzida a 16 bits ainda permite representar a dimensão adequada da quantidade de RAM num *laptop*, em bytes (2^{32}).

5. Imediatamente após a conclusão da execução duma instrução, considere a execução duma nova instrução do IA-32 (*little endian*), representada em *assembly* por: `andl %ebx, -12(%ebp)`.

A instrução em binário está em memória e ocupa 3 células de memória: as 2 primeiras com o código da operação e a especificação do local onde estão os operandos, com os valores `0x83` e `0x7c`, enquanto a 3ª contém a constante -12.

Considere os seguintes conteúdos de registos:

`%ebx 0x1, %esp 0x8c28efc, %eip 0x8c20484, %ebp 0x8c28f0c.`

Considere ainda este conteúdo das 16^{as} células no topo da *stack* (endereços por ordem crescente, em hexadecimal): `00, 00, 00, 00, 83, 00, 01, 00, c4, 02, 80, 00, 40, ff, ff, ff, 2c, 8f, c2, 08`

- a) Por ordem cronológica, toda a informação que circula apenas no barramento de endereços é a seguinte: `0x08c20484, 0x00000001, 0x08c28f0c, 0x8c28f00`
- b) Após a execução desta instrução, apenas o registo `%ebp` foi alterado e a única célula de memória que foi modificada foi no endereço `0x8c28f00`: alterada para `01`.