

Este exame tem 6 questões em 4 páginas, num total de 200 pontos. Fundamentar todas as respostas. Usar uma folha simples para juntar as respostas às perguntas 1, 2 e 3.

Escolha múltipla

1. Indicar na folha de prova a única resposta correta a cada uma das seguintes alíneas.

[8] (a) Considerar o seguinte fragmento de código *assembly*:

```
.data
v1 BYTE 55H
v2 BYTE F0H
.code
xor al, al
or al, v1
xor al, v2
```

O valor final de AL é:

A. 55H B. 50H C. 0F5H D. 0A5H

[8] (b) Considerar a seguinte sub-rotina.

```
func PROC x:sdword, y:sdword
LOCAL tmp: sdword
...
func ENDP
```

Qual das seguintes instruções é inválida?

A. `mov tmp, y` B. `mov eax, y` C. `mov tmp, ecx` D. `lea eax, tmp`

[8] (c) O disco magnético de um computador é substituído por outro com maior velocidade de rotação, sendo as restantes características iguais. Qual dos seguintes fatores é afetado negativamente?

A. débito B. tempo de resposta
C. consumo de energia D. capacidade de armazenamento

[8] (d) Qual das seguintes afirmações sobre DMA (acesso direto a memória) é falsa?

A. A utilização de DMA requer um controlador específico para esse fim.
B. Transferências por DMA não requerem o uso de interrupções.
C. A utilização de DMA é mais complicada que a realização de varrimentos (*polling*).
D. A utilização de DMA é indicada para dispositivos de alta velocidade.

[8] (e) Um sistema RAID-6 usa discos de 0,5 TB. Pretende-se que o sistema tenha uma capacidade útil mínima de 19 TB. Por razões de fiabilidade, cada grupo não deve ter mais de 6 discos no total. Qual é o número mínimo de discos necessário?

A. 65 B. 60 C. 76 D. 48

Sistemas de entrada/saída

- [20] 2. Um disco magnético possui 64 setores de 4kB por pista e a sua velocidade de rotação é 10000 RPM. Nas condições mais desfavoráveis um ficheiro de 40kB demora 121ms a ser lido e 13ms nas condições mais favoráveis. Calcule o tempo médio de busca e a taxa de transferência desse disco.

Nota: Comece por estabelecer as expressões do tempo de leitura do ficheiro, em função dos parâmetros pedidos, para as duas situações referidas. [Considere kB = 10^3 B, MB = 10^6 B.]

3. Um computador possui um CPU capaz de executar 1000 MIPS. O barramento principal desse computador (FSB) tem uma largura de banda de 800 MB/s. O sistema tem 4 controladores de disco. Cada controlador é capaz de transferir até 200 MB/s e de controlar até 4 discos em simultâneo. O sistema possui um disco magnético de 500 GB instalado em cada controlador, com setores de 4kB, taxa de transmissão de 80 MB/s, tempo de busca médio de 9ms e latência de rotação de 0,95ms. O acesso a um setor gasta 10000 instruções do CPU. [Considere kB = 10^3 B, MB = 10^6 B.]
- [10] (a) Indique qual dos componentes fixos (CPU, FSB, controladores de disco) está a limitar o desempenho global do sistema.
- [10] (b) Indique se é possível quadruplicar a capacidade de armazenamento do sistema continuando a recorrer a discos de 500 GB e taxa de transmissão 80 MB/s (i) semelhantes aos já instalados, (ii) em tecnologia SSD.

Programação

4. Pretende-se imprimir, na forma de gráfico de barras horizontal, o número de ocorrências de cada dígito de uma sequência aleatória de dígitos ASCII de tamanho máximo 50 dígitos e terminada por 0. Exemplo (38 dígitos ASCII):

```
31099581816517957956980691175735181857
=====
9: =====
8: =====
7: =====
6: =====
5: =====
4:
3: ==
2:
1: =====
0: ==
=====
```

- [20] (a) Escrever a rotina `freqdig PROTO Buf:ptr byte, Dig:byte` que conta o número de ocorrências do dígito `Dig` na sequência apontada por `Buf`.
- [10] (b) Escrever a rotina `pbar PROTO N:byte` que imprime uma barra de sinais '=' de tamanho `N` ($0 \leq N \leq 50$), recorrendo à variável global pré-inicializada `barra`.

`barra BYTE 53 dup('='),0`

Nota: repare que para controlar o tamanho da barra a imprimir basta colocar o terminador de *string* na posição conveniente da variável `barra`.

- [10] (c) Completar o programa principal, nos dois locais assinalados com "...", que imprime o gráfico de barras pretendido com o formato indicado no exemplo.

```
include mpcp.inc
;; sub-rotinas definidas noutro ficheiro
freqdig proto Buf:ptr byte, Dig:byte
pbar      proto N:byte
      .data
texto byte  '31099581816517957956980691175735181857',0
CRLF  byte  10,13,0
barra byte  53 dup('='),0
      ...

      .code
main: invoke printf, offset texto
      invoke printf, offset CRLF
      invoke printf, offset barra
      invoke printf, offset CRLF
      ...
      invoke printf, offset barra
      invoke _getch
      invoke ExitProcess,0
end main
```

5. O movimento de uma partícula em função do tempo t é definido pela função

$$s(t) = \begin{cases} 2 + t \times \sin(\pi \times t/2) & \text{se } 0 \leq t < 2 \\ 4 \times \frac{\cos(t-2)}{t} & \text{se } t \geq 2 \end{cases}$$

- [20] (a) Implemente a sub-rotina `dist` que calcula o valor de $s(t)$.

O protótipo da sub-rotina é: `dist proto t:real8`

- [20] (b) A velocidade média de uma partícula no intervalo de tempo $[t_1, t_2]$ define-se como $\bar{v}(t) = \frac{s(t_2) - s(t_1)}{t_2 - t_1}$.

Implemente a sub-rotina `velM` que calcula o valor de $\bar{v}(t)$. O protótipo da sub-rotina é: `velM proto t1:real8, t2:real8`

Análise de código

6. Considere o seguinte programa constituído por dois ficheiros (`prog.cpp` e `rotinas.asm`).

```
// Ficheiro prog.cpp
extern "C" void newSeq(int *s1, int *s2, int *s, unsigned int nval);
int SEQ1[6]={4, -9, 3, 8, 5, 4},
    SEQ2[6]={2, -1, 3, 0, 1, -3}, S[6];

main ()
{
    newSeq(SEQ1, SEQ2, S, 6);
    for (unsigned int i=0; i<6; i++)
        std::cout << S[i] << ' ';
    return 0;
}
```

```

1  ;; Ficheiro rotinas.asm
2  sumN proc uses ebx esi pt:ptr sdword, N:sdword
3      mov     esi, pt
4      mov     eax, [esi]
5      cmp     N, 0
6      je      L3
7      mov     ecx, N
8      .if     N < 0
9          neg     ecx
10 L1:      sub     esi, 4
11          add     eax, [esi]
12          loop   L1
13      .else
14 L2:      add     esi, 4
15          add     eax, [esi]
16          loop   L2
17      .endif
18 L3:      ret
19 sumN endp
20
21 newSeq proc uses esi edi ebx pt1:ptr sdword, pt2:ptr sdword,
22                                     pt:ptr sdword, N:dword
23     mov     ecx, N
24     mov     esi, pt1
25     mov     ebx, pt2
26     mov     edi, pt
27     .while  ecx > 0
28         push  ecx
29         invoke sumN, esi, sdword ptr [ebx]
30         mov   [edi], eax
31         add   esi, 4
32         add   ebx, 4
33         add   edi, 4
34         pop   ecx
35         dec   ecx
36     .endw
37     ret
38 newSeq endp

```

- [10] (a) Descreva a função realizada pela sub-rotina `sumN`.
- [10] (b) Indique e explique o que é apresentado no monitor quando o programa é executado.
- [10] (c) Admita que `SEQ1` passa a ter um sétimo elemento igual a -1. Nestas condições, sabendo que após a sétima execução de `sumN` resulta `EAX=8`, determine qual deverá ser o sétimo elemento de `SEQ2`.
- [10] (d) Da primeira vez que o prólogo de `sumN` é executado resulta o conteúdo da pilha apresentado na tabela. Determine os endereços, inicial e final, das posições de memória ocupadas pela sequência `SEQ1` declarada em `main()`.

Endereço (hex.)	Conteúdo (hex.)
004CFA0C	004CFA2C
004CFA08	70881022
004CFA04	7088100A

Fim do enunciado.