

Esta prova tem 7 questões, num total de 200 pontos. Fundamente todas as respostas.

## Análise de código

1. Considere o seguinte programa constituído por dois ficheiros.

```
#include <iostream>
typedef unsigned int uint;
extern "C" void asmf(uint *seq1, uint n,
                    uint *seq2, uint *seq3, uint m);
uint valores[6] = {15, 11, 12, 18, 30, 21};
uint msk[2] = {3, 16};
uint res[2] = {0, 0};
int main()
{
    asmf(valores, 6, msk, res, 2);
    std::cout << res[0] << ' ' << res[1] << std::endl;
}
```

```
1  aux PROC USES ebx s:PTR DWORD, n:DWORD, msk:DWORD
2      xor    eax, eax
3      mov    ecx, n
4      jecxz  fim
5      mov    edx, s
6  @@:    mov    ebx, [edx]
7      and    ebx, msk
8      .IF    ebx != 0
9          inc    eax
10     .ENDIF
11     add    edx, 4
12     loop   @B
13 fim:    ret
14 aux    ENDP
15
16 asmf PROC USES esi edi ebx seq1:PTR DWORD, n:DWORD,
17         seq2:PTR DWORD, seq3:PTR DWORD, m:DWORD
18     xor    ebx, ebx
19     mov    esi, seq2
20     mov    edi, seq3
21     .WHILE ebx < m
22         invoke aux, seq1, n, [esi+4*ebx]
23         mov    [edi+4*ebx], eax
24         inc    ebx
25     .ENDW
26     ret
27 asmf    ENDP
```

- [10] (a) Considere a primeira invocação da sub-rotina `aux` realizada a partir da linha 22. Assuma que `seq1 = 00410A00H`.

Indique os valores dos registos `EAX`, `EBX`, `ECX` e `EDX` após a execução das instruções das linhas 6 e 12 para cada uma das três primeiras iterações do ciclo.

**Resposta:** Iteração 1:

`EAX = 0`, `EBX = 15` , `ECX=6`, `EDX = 00410A00H`

`EAX = 1`, `EBX = 3` , `ECX=5`, `EDX = 00410A04H`

Iteração 2:

`EAX = 1`, `EBX = 11` , `ECX=5`, `EDX = 00410A04H`

`EAX = 2`, `EBX = 3` , `ECX=4`, `EDX = 00410A08H`

Iteração 3:

`EAX = 2`, `EBX = 12` , `ECX= 4`, `EDX = 00410A08H`

`EAX = 2`, `EBX = 0` , `ECX =3`, `EDX = 00410A0CH`

- [10] (b) Durante a execução do programa, quantas iterações do ciclo das linhas 6 a 12 são realizadas?

**Resposta:** A sub-rotina é invocada duas vezes. Para cada invocação, o ciclo tem  $n$  iterações (valor inicial do registo `ECX`). Como  $n = 6$ , são executadas 12 iterações ao todo.

- [10] (c) Indique o que é apresentado no monitor e explique a tarefa realizada pelo programa.

**Resposta:** No final da execução, a mensagem que surge no monitor é a seguinte:  
5 3.

Para cada valor de `SEQ2`, o programa determina quantos valores de `SEQ1` têm pelo menos um bit a 1 em comum com esse valor. O número de valores é determinada pela sub-rotina `aux`. Esta sub-rotina percorre uma sequência, fazendo o E-lógico entre `msk` e os elementos da sequência. O resultado dessa operação é diferente de zero, sempre que os dois operandos tiverem pelo menos um bit a 1 em comum.

A sub-rotina `asmf` assegura que este processo é repetido para todos os valores da sequência `msk` (correspondendo ao argumento `seq2`), guardando os respetivos resultados na sequência `res` (correspondendo ao argumento `seq3`).

- (d) Considere a primeira invocação da sub-rotina `aux` realizada durante a execução do programa. Após a execução da instrução da linha 3, tem-se `ESP = 05000A00H` e o estado da pilha é o seguinte (valores em hexadecimal):

Endereço	Conteúdo
05000A14	00000003
05000A10	??
05000A0C	00410A00
05000A08	00400D00
05000A04	050009DC
05000A00	00000000

- [5] i. Qual é o valor do elemento da pilha situado no endereço de memória 05000A10H?

**Resposta:** Os valores residentes na pilha são (a partir do topo lógico da pilha, i.e., de baixo para cima): EBX (devido à diretiva USES da sub-rotina aux), EBP anterior, endereço de retorno, s, n, msk[0]. Logo, o valor pedido é o valor do parâmetro n = 00000006H = 6.

- [5] ii. Qual é a posição em memória do código da instrução da linha 23?

**Resposta:** A instrução da linha 23 é aquela que será executada quando a sub-rotina aux terminar a execução, i.e., a instrução está na posição de memória cujo endereço está na pilha (como endereço de retorno). Esse endereço é o terceiro elemento a partir do topo, ou seja 00400D00H.

## Sistemas de entrada/saída

[Nota: Para simplificar os cálculos assuma kB=10<sup>3</sup> B, MB=10<sup>6</sup> B, GB=10<sup>9</sup> B e TB=10<sup>12</sup> B]

2. Um controlador de discos magnéticos pode gerir vários discos iguais. Cada disco suporta uma largura de banda de 22 MB/s usando blocos de 11 kB. Os discos possuem uma velocidade de rotação de 9000 RPM e um tempo de busca igual ao dobro do tempo médio de rotação. O tempo introduzido pelo controlador num acesso a disco é 0,5 ms.

- [10] (a) Calcule o tempo médio de acesso a disco.

**Resposta:** O tempo médio de acesso a disco é dado por

$$t_{\text{acesso}} = t_{\text{busca}} + t_{\text{rotação}} + t_{\text{transferência}} + t_{\text{controlador}}$$

De acordo com os dados fornecidos, resulta:

$$t_{\text{busca}} = 2 \times t_{\text{rotação}}$$

$$t_{\text{rotação}} = \frac{1/2}{9000/60} = 10/3 \text{ ms}$$

$$t_{\text{transferência}} = \frac{\text{tamanho do bloco}}{\text{largura de banda}} = \frac{11 \text{ kB}}{22 \text{ MB/s}} = 0,5 \text{ ms}$$

Logo,

$$t_{\text{acesso}} = (2 + 1) \times 10/3 + 0,5 + 0,5 = 11 \text{ ms}$$

- [10] (b) Determine a taxa de transferência média entre controlador e disco. (Nota: caso não tenha resolvido a alínea anterior, assuma que o tempo de acesso a disco é 10 ms.)

**Resposta:**

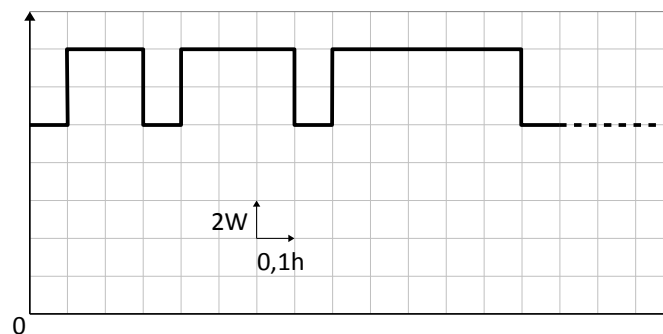
$$\text{taxa de transferência média} = \frac{\text{tamanho do bloco}}{\text{tempo de acesso}} = \frac{11 \text{ kB}}{11 \text{ ms}} = 1 \text{ MB/s}$$

- [10] (c) Assuma que o acesso a disco passa a ser feito em blocos com o dobro da dimensão. Mostre que a taxa de transferência média não passa ao dobro.

**Resposta:**

Se os blocos passam a ter o dobro da dimensão, a taxa de transferência média só passará para o dobro se o tempo de acesso continuar com o mesmo valor. Porém, tal não acontece porque a alteração do tamanho dos blocos afeta o tempo de acesso.

- [20] 3. Num computador portátil a rede sem fios gasta 4 W quando está ligada e o resto do hardware gasta 10 W. A figura representa o perfil de utilização desse computador.



Sabendo que a bateria do computador tem uma capacidade de 65 W h, determine a sua autonomia total assumindo que, no tempo restante, a rede estará sempre desligada.

Notas: 1 J = 1 W × 1 s, 1 W h = 3600 J.

**Resposta:**

Energia gasta nas primeiras 1,3 h:  $1 \text{ h} \times 14 \text{ W} + 0,3 \text{ h} \times 10 \text{ W} = 14 \text{ W h} + 3 \text{ W h} = 17 \text{ W h}$

Energia restante:  $65 - 17 = 48 \text{ W h}$

Autonomia restante sem rede:  $\frac{48}{10} = 4,8 \text{ h}$

Autonomia total:  $1,3 + 4,8 = 6,1 \text{ h}$

---

## Programação

---

4. A partir do momento em que um avião se prepara para descolar, a sua velocidade é registada automaticamente, de segundo a segundo, na sequência `velsec`, residente na memória de uma das suas caixas negras. Em cada registo guarda-se um número inteiro que representa o valor da velocidade instantânea, em m/s.

- [10] (a) Assumindo que se deve poder armazenar velocidades até 900 km/h (250 m/s), durante 8 h (28800 s), declare a variável `velsec` de forma a minimizar o espaço de memória ocupado e inicializar todos os seus elementos com o valor zero.

**Resposta:**

Os valores a guardar variam entre 0 e 250 por isso uma variável do tipo `byte` (8 bits sem sinal) será a mais eficiente em termos de minimização do espaço de memória ocupado:

```
.data
velsec byte 28800 dup(0)          ; 28800 valores inicializados a zero
```

- [20] (b) No fim de uma viagem pretende-se saber que percentagem de tempo o avião viajou acima de uma dada velocidade `vx`. Escreva a rotina `supvx` que retorna essa percentagem (nº inteiro entre 0 e 100). A rotina tem o seguinte protótipo:

`supvx proto vx:byte`

**Resposta:**

```
supvx proc uses esi ebx vx:byte
    mov esi,offset velsec      ; aponta primeiro elemento
    mov ecx,lengthof velsec    ; numero maximo de amostras
    mov ah,vx                  ; limiar de velocidade em AH
    xor ebx,ebx                ; segundos de viagem
    xor edx,edx                ; segundos acima do limiar

@@: lodsb                     ; ler velocidade instantanea
    cmp al,0                   ; se chegou ao fim da viagem
    jz fim                     ; deve acabar
    inc ebx                    ; mais um segundo de viagem...
    .if(al>ah)
        inc edx                ; mais um segundo acima do limiar...
    .endif
    loop @b

fim: mov eax,100               ; converte para percentagem
    mul edx                    ; EAX = 100 * EDX
    xor edx,edx                ; prepara divisao
    div ebx                    ; divide pela duracao da viagem
    ret                        ; resultado em EAX
supvx endp
```

- [20] 5. A expressão seguinte permite calcular o valor da função de Weierstrass:

$$W(x) = \sum_{i=0}^{+\infty} a^i \cos(b^i \pi x) \quad (a, b > 0)$$

Escreva a sub-rotina FW, que calcula o valor aproximado desta função considerando  $N$  parcelas ( $i = 0, \dots, N - 1$ ). Considere que o protótipo da sub-rotina é:

FW PROTO N:DWORD, a:REAL8, b:REAL8, x:REAL8

### Resposta:

```
FW PROC N:DWORD,a:REAL8,b:REAL8,x:REAL8
    local an:REAL8
    local bn:REAL8
    local pix:REAL8

    mov     ecx, N
    fld1
    fst     an
    fstp    bn
    fldpi
    fmul    x
    fstp    pix
    fldz
    ; Soma inicial

    fmul    pix
    fcos
    fmul    an
    fadd
    ; Atualiza somatorio

    fld     an
    fmul    a
    fstp    an
    ; an = an-1 * a

    fld     bn
    fmul    b
    fstp    bn
    ; bn = bn-1 * b

    loop    @B
    ret

FW ENDP

@@: fld     bn
```

6. A sub-rotina seguinte calcula uma função  $z(x)$ , encontrando-se incompleta.

```
FUNZ PROC x:REAL8
    fldz
    fld     x
    fcomip  st(0), st(1)
    fstp    st(0)
    ja      maior
    fld1
    fld1
    fadd
    fld     x
    fmul    x
    fcos
    fsub
    fdivr   x
    jmp     fim
maior: ...
```

```

    ...
fim:   ret
FUNZ  ENDP

```

Da função sabe-se que

$$z(x) = \begin{cases} \dots & \text{se } \dots \\ -\frac{1}{x+\frac{1}{x}} & \text{se } x > 0 \end{cases}$$

- [10] (a) Indique os elementos em falta na expressão de  $z(x)$ .

**Resposta:**  $\frac{x}{2 - \cos x^2}$  se  $x \leq 0$

- [10] (b) Complete a sub-rotina indicando *apenas* o código em falta.

**Resposta:**

```

maior: fld1
      fdiv  x
      fadd  x
      fld1
      fdivr
      fchs

```

## Escolha múltipla

7. Para as alíneas seguintes, indique a única resposta correta.

- [8] (a) Qual é o valor de EDX após a execução do seguinte fragmento?

```

.data
seq SDWORD 10, 20, 30, 40, 50, 60, 70, 80, 90
.code
mov ESI, OFFSET seq
mov EDX, [ESI+8]

```

A. 10   B. 20   **C. 30**   D. Nenhum dos anteriores

**Resposta:** Opção C.

- [8] (b) Considere um multiprocessador de memória partilhada do tipo NUMA. Indique a afirmação verdadeira.

A. O desempenho de um programa paralelo é independente da posição dos dados em memória.

**B. Qualquer processador pode aceder a qualquer posição da memória partilhada.**

C. Um modelo com  $2N$  processadores garante um desempenho 2 vezes superior ao de um modelo com  $N$  processadores.

D. Nenhuma das anteriores afirmações é verdadeira.

**Resposta:** Opção B.

- [8] (c) Um sistema RAID-5 usa 32 discos de 1TB e tem uma capacidade útil de 24TB. Quantos grupos de proteção tem o sistema?

A. 1   B. 2   C. 4   **D. 8**

**Resposta:** Opção D.

- [8] (d) Qual é o valor de ECX após a execução do seguinte fragmento?

```
.data
vals BYTE 10, 9, 8, 7, 6
.code
cld
mov     EDI, OFFSET vals
mov     ECX, LENGTHOF vals
mov     AL, 8
repne   scasb
```

A. 4   B. 3   **C. 2**   D. 0

**Resposta:** Opção C.

- [8] (e) Qual é o valor do registo AH após a execução do seguinte fragmento?

```
.data
val    BYTE 10H
.code
mov     AL, F5H
mul     val
```

**A. 0FH**   B. 05H   C. 1FH   D. 15H

**Resposta:** Opção A.

Fim do enunciado.