

Este exame tem 7 questões, num total de 200 pontos. Responda em folhas separadas a cada um dos seguintes conjuntos de problemas: (1 e 2), (3 e 4), (5 e 6). O problema 7 deve ser respondido na folha de enunciado.

1. Considerar o seguinte programa composto por dois ficheiros.

```
#include <iostream>
using namespace std;
extern "C" unsigned int filtro(unsigned int *v, unsigned int n);
const int nval = 8;
unsigned int valores[nval] = {0, 6, 9, 12, 16, 10, 16, 0};
int main()
{
    int x = filtro(valores, nval);
    cout << "R=" << x << " valores: ";
    for (int i = 0; i < nval; i++) cout << valores[i] << ' ';
    cout << endl;
    return 0;
}
```

```
1 include mpcp.inc
2 .code
3 filtro PROC C USES edi ebx esi v:PTR DWORD, n:DWORD
4     LOCAL tres: DWORD
5         mov     tres, 3
6         mov     edi, v
7         mov     ebx, [edi]
8         mov     ecx, n
9         sub     ecx, 2
10        add     edi, 4
11        xor     esi, esi
12 ciclo:
13        xor     edx, edx
14        mov     eax, [edi]
15        add     eax, ebx
16        mov     ebx, [edi]
17        add     eax, [edi+4]
18        div     tres
19        .IF     edx == 2
20            inc     eax
21        .ENDIF
22        .IF     eax != [edi]
23            mov     [edi], eax
24            inc     esi
25        .ENDIF
26        add     edi, 4
27        loop    ciclo
28        mov     eax, esi
29        ret
30 filtro ENDP
31 END
```

Quando executado, o programa apresenta no monitor a seguinte mensagem:

R=4 valores: 0 5 9 12 13 14 9 0

**Nota:** Justificar todas as respostas.

- [10] (a) Quantas vezes é executado o corpo do ciclo das linhas 12–27?

**Resposta:** A repetição do ciclo é controlada pela instrução `loop` da linha 27, que usa o valor de `ECX` para determinar se deve ser executada uma nova iteração. No corpo do ciclo não existe nenhuma instrução de salto para fora do ciclo, nem instruções que alterem o valor do registo `ECX`. Portanto, o valor de `ECX` antes do ciclo vai determinar o número de iterações. Este valor é determinado na linha 9 como sendo  $n - 2 = 6$ . Portanto, o corpo do ciclo é repetido 6 vezes.

- [10] (b) Quantas vezes é executada a instrução da linha 20 e qual é a sua finalidade (para o algoritmo implementado pela sub-rotina)?

**Resposta:** O ciclo calcula a média de 3 valores, pelo que realiza uma divisão por 3. Como se trata de números inteiros, o resto da divisão fica em `EDX` e o quociente (inteiro) em `EAX`. Para que o valor inteiro em `EAX` seja o valor inteiro mais próximo do resultado real da divisão, é preciso examinar o resto. Neste caso, o resto pode ser 0, 1 ou 2. Se for 0 ou 1, então o valor de `EAX` já é o valor correto. Se o resto for 2 (o que corresponde a um número com parte fracionária de  $2/3 \approx 0,667$ ), o valor correto é obtido adicionando uma unidade ao valor de `EAX`. O incremento condicional é realizado pelo bloco `.IF/.ENDIF` que contém a linha 20.

Para determinar o número de vezes que a instrução é executada, é necessário calcular as 6 médias e verificar quantas vezes a condição se verifica. Para os dados indicados no código C++, isso ocorre 2 vezes.

- [10] (c) Qual é o significado do valor retornado pela sub-rotina `filtro`?

**Resposta:** O resultado da sub-rotina é definido pelo valor final de `EAX`, que é o valor de `ESI` (linha 28) após o ciclo terminar. Inicialmente (antes do ciclo) o valor de `ESI` é 0; o valor é incrementado sempre que o teste da linha 22 é verdadeiro, isto é, quando o valor existente na posição referenciada por `ESI` é diferente do valor calculado. Portanto, `ESI` é usado como contador das alterações feitas aos elementos da sequência (pela instrução da linha 23). Logo, o resultado da sub-rotina é o número de valores da sequência que foram alterados.

- [10] (d) Considerando a sub-rotina `filtro` em geral (e não apenas neste programa), qual é a gama dos resultados que pode produzir? (Assumir que se tem sempre  $n \geq 3$ .)

**Resposta:** A sub-rotina retorna o número de elementos alterados. Como os dois extremos da sequência nunca são alterados, o valor máximo é  $n - 2$ . O valor mínimo é 0 (por exemplo, para uma sequência cujos elementos sejam todos iguais). Portanto, o resultado está sempre na gama  $[0; n-2]$ .

2. A tabela apresenta o conteúdo da pilha resultante da invocação de uma sub-rotina e da execução do respetivo prólogo (convenção STDCALL). Sobre a sub-rotina sabe-se que altera os registos EAX, EBX e ECX, e que não possui variáveis locais.

Endereço (hex.)	Conteúdo (hex.)
004FF10C	00000270
004FF108	000001A5
004FF104	033F1000
004FF100	00AB3A9F
004FF0FC	004FF11C
004FF0F8	000000FF

- [10] (a) Indique o que representa cada item do conteúdo.

**Resposta:**

Da informação sobre os registos alterados pela sub-rotina conclui-se que EBX é o único registo a preservar (convenção STDCALL). Portanto, após a execução do prólogo, o topo da pilha representa o valor de EBX. No endereço seguinte (004FF0FCh) encontra-se o valor de EBP (a primeira instrução do prólogo é sempre `push ebp`). Antes do prólogo, ou seja, na invocação da sub-rotina, o último valor guardado na pilha é o endereço de retorno da sub-rotina. Os conteúdos indicados nas três primeiras linhas da tabela referem-se portanto aos argumentos (3) com que ocorreu a invocação.

Endereço (hex.)	Conteúdo (hex.)	Significado
004FF10C	00000270	3º argumento da invocação da sub-rotina
004FF108	000001A5	2º argumento da invocação da sub-rotina
004FF104	033F1000	1º argumento da invocação da sub-rotina
004FF100	00AB3A9F	Endereço de retorno da sub-rotina
004FF0FC	004FF11C	Valor de EBP
004FF0F8	000000FF	Valor de EBX

- [10] (b) Apresente o epílogo da sub-rotina.

**Resposta:**

```
pop    ebx
leave
ret    12 ; Número de bytes ocupados pelos argumentos da sub-rotina
```

3. Um sistema possui um CPU que opera a 4 GHz. Este sistema possui ainda um disco que transfere dados para o processador em grupos de 4 palavras (2 bytes cada) e tem uma taxa de transferência de dados de 40 MB/s. [Considere  $\text{kB} = 10^3 \text{ B}$ ,  $\text{MB} = 10^6 \text{ B}$ .]

- [10] (a) Se pretendermos utilizar *polling* como técnica de gestão de periféricos e sabendo que uma operação de *polling* consome 400 ciclos de relógio, qual é a fração de tempo de CPU consumida?

**Resposta:**

Dados transferidos por acesso:

$$4 \times 2\text{B} = 8\text{B}$$

Acessos por segundo:

$$\frac{40 \text{ MB/s}}{8 \text{ B}} = \frac{40 \times 10^6}{8} = 5 \times 10^6$$

Então, o número de ciclos consumidos pela operação é de:

$$5 \times 10^6 \times 400 = 2 \times 10^9$$

Tento em consideração o número de ciclos consumidos por segundo, a percentagem média de tempo de CPU gasto é:

$$\frac{2 \times 10^9}{4 \times 10^9} = 0,5 = 50 \%$$

- [5] (b) Se em vez de *polling* fosse utilizada a técnica de interrupções, qual seria a fração de tempo de CPU consumida? Admita que o *overhead* de cada transferência, incluindo o atendimento da interrupção, é de 600 ciclos de relógio e que o disco está sempre potencialmente ocupado.

**Resposta:**

Dados transferidos por acesso:

$$4 \times 2\text{B} = 8 \text{ B}$$

Acessos por segundo:

$$\frac{40 \text{ MB/s}}{8 \text{ B}} = \frac{40 \times 10^6}{8} = 5 \times 10^6$$

Então, o número de ciclos consumidos pela operação é de:

$$5 \times 10^6 \times 6 \times 10^2 = 3 \times 10^9$$

Tento em consideração o número de ciclos consumidos por segundo, a percentagem média de tempo de CPU gasto é:

$$\frac{3 \times 10^9}{4 \times 10^9} = 0,75 = 75 \%$$

- [5] (c) Comparando os resultados das alíneas anteriores, em que situações é que a técnica de interrupções pode ser vantajosa?

**Resposta:**

No segundo cenário, o custo (em ciclos de relógio) do *overhead* de cada transferência, incluindo o atendimento da interrupção, é superior ao custo de uma operação de *polling*. A técnica de interrupções é mais vantajosa quando o periférico não está sempre potencialmente ocupado. Para este caso concreto, a técnica de interrupções é mais vantajosa quando o disco está ocupado menos de 66,7 % do tempo.

$$\frac{X \times 3 \times 10^9}{4 \times 10^9} = 0,5$$

$$X \times \frac{3}{4} = 0,5$$

$$X = \frac{0,5}{0,75} = 66\%$$

- [20] 4. A sub-rotina CalculaNF recebe as notas da avaliação distribuída (NAD) e do exame (NE) e calcula a nota final de um estudante (valor inteiro entre 0 e 20) aplicando a seguinte expressão:

$$\max(0,4 \times \text{NAD} + 0,6 \times \text{NE}; \text{NE})$$

em que NAD é a nota da avaliação distribuída e NE é a nota do exame. Apresentar a implementação da sub-rotina assumindo as seguintes definições:

PAD **REAL4** 0.4  
PE **REAL4** 0.6

O protótipo da sub-rotina é: CalculaNF PROTO nad:real4, ne:real4

### Resposta:

```
CalculaNF PROC notaad:real4, notae:real4
    movss    xmm1, notaad
    movss    xmm2, notae
    mulss    xmm1, PAD
    mulss    xmm2, PE
    addss    xmm1, xmm2
    maxss    xmm1, notae
    cvtss2si eax, xmm1
    ret
CalculaNF ENDP
```

- [30] 5. Apresente a sub-rotina sint que calcula a função  $y = f(x, n) = \frac{x^n}{n!}$  com x real e n inteiro positivo inferior a 12. O protótipo da sub-rotina é:

sint PROTO argx:real4, argn:dword, argy:ptr real4

### Resposta:

```
sint proc argx:real4, argn:dword, argy:ptr real4
    mov     eax, 1
    cvtsi2ss xmm0, eax
    movss   xmm1, argx
    mov     ecx, argn
@@: mulss  xmm0, xmm1
    mul     ecx
    loop    @b
    cvtsi2ss xmm1, eax
    divss   xmm0, xmm1
    mov     eax, argy
```

```

movss    real4 ptr[ebx], xmm0
ret
sint endp

```

6. Cada uma das seguintes questões tem apenas uma resposta certa. Indique as respostas corretas **na folha de resposta** (e não na folha do enunciado).

- [5] (a) Considere a instrução **comiss xmm0, xmm1**. Para consultar o resultado da comparação:
- A. deve usar-se sempre uma operação condicional para comparação de valores com sinal;
  - B. a operação condicional a usar depende dos valores terem ou não sinal;
  - C. deve usar-se uma operação condicional para comparação de valores sem sinal;**
  - D. é necessário usar operações lógicas SIMD para dados empacotados.
- [5] (b) O conteúdo inicial de `xmm0` é composto por 4 valores {S3, S2, S1, S0} do tipo REAL4. Pretende-se modificar o conteúdo do registo conforme indicado na figura.



Indique a opção que executa esta alteração.

- A. `shufps xmm0, xmm0, 93h`
  - B. `rol xmm0, 1`
  - C. `shufpd xmm0, xmm0, 1`  
`shufps xmm0, xmm0, 3Ah`
  - D. `insertps xmm0, xmm0, C0h`  
`shufps xmm0, xmm0, 39h`
- [5] (c) Considerar a sub-rotina cujo cabeçalho é:
- ```

rotX PROC C uses ebx arg1:SWORD, arg2:SWORD

```
- A limpeza da pilha será assegurada:
- A. pela sub-rotina, no seu prólogo;
  - B. pela sub-rotina, no seu epílogo;
  - C. pelo programa que chamar a sub-rotina;**
  - D. pela sub-rotina, no seu corpo principal.
- [5] (d) Assumir que, inicialmente, `EAX=00003F26H` e `EBX=55550000H`. Qual é o valor de `EAX` após execução de

```

xor     eax, 0FFFFFFAAH
cmovns  eax, ebx
xor     eax, 0FFFFFFAAH

```

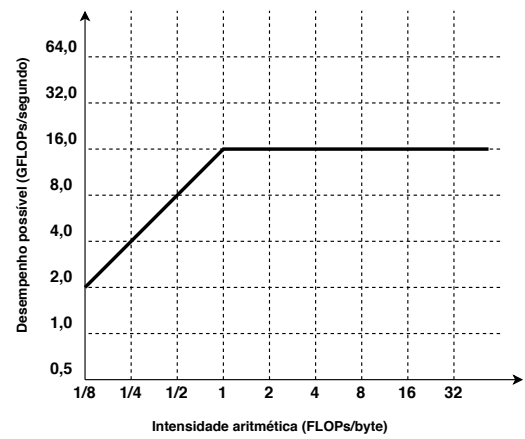
- A. 0AAAAAAAH   B. 5555AAAAH   C. 0FFFFC0D9H   **D. 00003F26H**

- [5] (e) Um vetor tem 40 elementos. Pretende-se elevar ao quadrado todos os elementos e calcular a raiz quadrada da sua soma. Assuma que todas as operações de cálculo demoram o mesmo tempo. Usando um multiprocessador de memória partilhada equipado com 10 processadores, qual é o aumento de desempenho (*speed-up*) ideal obtido em relação a um único processador?
- A. 10   **B. 80/12 ≈ 6,67**   C. 44/12 ≈ 3,67   D. 44/11 = 4
- [5] (f) As características de dois programas de cálculo científico estão indicadas na tabela. Os dois programas são executados num multiprocessador caracterizado pelo diagrama indicado.

| Programa                   | P1  | P2  |
|----------------------------|-----|-----|
| Intensidade aritmética     | 1/2 | 4   |
| Esforço de cálculo (GFLOP) | 800 | 320 |

Qual é o tempo total de execução dos dois programas?

- A. 90 s    **B. 120 s**    C. 140 s    D. 70 s

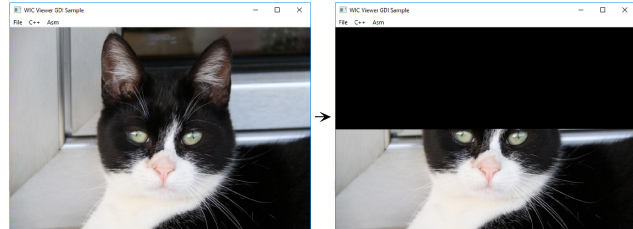


(continua na pág. 8)

Nome (legível): \_\_\_\_\_

7. Uma imagem é armazenada em memória como uma sequência de píxeis resultante da sucessão de linhas que compõem a imagem. A informação de cada píxel (componentes R, G e B, e transparência) ocupa 4 bytes.

- [20] (a) Pretende-se alterar uma imagem colocando a preto todos os píxeis da metade superior da imagem, tal como exemplificado na figura. Implemente a sub-rotina FXP que execute esta alteração.



O protótipo da sub-rotina é: FXP PROC pt: ptr byte, ncol: dword, nlin: dword.

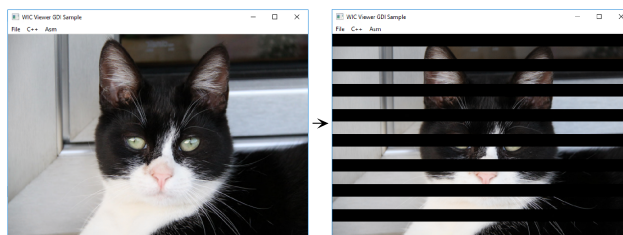
O endereço inicial da imagem é dado por pt, representando ncol e nlin o número de píxeis por linha (largura) e o número de linhas (altura), respetivamente. Assuma que nlin é par.

**Resposta:**

```
FXP PROC uses edi pt: ptr byte, ncol: dword, nlin: dword
    mov     eax, ncol
    mul     nlin      ; EAX = número de píxeis da imagem
    mov     ecx, eax
    shr     ecx, 1     ; ECX = número de píxeis a alterar (= número de iterações)
    mov     edi, pt
@@: mov     dword ptr [edi], 0    ; Torna o píxel preto
    add     edi, 4
    loop    @B
    ret
FXP ENDP
```



- [20] (b) Pretende-se alterar uma imagem dividindo-a em 8 faixas horizontais, colocando a preto a metade superior de cada uma delas, conforme exemplificado na figura. Assuma que a altura da imagem (número de linhas) é múltipla de 8.



Complete a sub-rotina seguinte, FXP8, de modo a executar a alteração descrita recorrendo à sub-rotina FXP.

FXP8 PROC uses edi pixels: ptr byte, largura: dword, altura:dword

```

; pixels = endereço inicial da imagem
; largura= número de píxeis por linha
; altura = número de linhas

```

local nlf:dword ; Número de linhas por faixa

local nbfc:dword ; Número de bytes por faixa

```

mov     ecx, 8          ; Número de faixas a alterar
mov     edx, altura
shr     edx, 3
mov     nlf, edx
mov     eax, largura
shl     eax, 2          ; Número de bytes por linha
mul     edx             ; EAX = número de bytes por faixa
mov     nbfc, eax
mov     edi, pixels
@@: push ecx
    invoke FXP, edi, largura, nlf
    pop     ecx
    add     edi, nbfc
    loop   @@
    ret
FXP8 ENDP

```

### Resposta:

Obs.:

- Em alternativa à instrução shl poderia ser usada a instrução sal. Estas instruções originam o mesmo resultado independentemente de se fazer um deslocamento lógico ou deslocamento aritmético para a esquerda.
- Quando a instrução de multiplicação é executada o valor de edx e de nlf são iguais. Assim, como alternativa ao operando indicado na solução pode considerar-se nlf.

Fim.