

Nome: \_\_\_\_\_ Nº de estudante: \_\_\_\_\_

**Atenção:** Este teste tem 12 questões em 4 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 × na grelha seguinte.

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

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

Pontos: \_\_\_\_\_ / 140

 [10] 1. O epílogo `leave/ret` 8 pertence a uma rotina que:

- A. tem 8 bytes de variáveis locais  
 B. tem 2 parâmetros de entrada  
 C. salva (e recupera) 2 registos de 32 bits  
 D. usa a convenção de chamada do C

[10] 2. Considere as seguintes declarações:

```
Rot1 proto stdcall v1:word, v2:dword
```

```
Rot2 proto C v1:word, v2:dword
```

e o seguinte fragmento de código:

```
push edx
push 0Ch
call 005643A3h
add esp,8
```

O fragmento de código corresponde à invocação:

- A. da rotina Rot2    B. da rotina Rot1    C. de nenhuma delas    D. de qualquer uma delas

 [10] 3. Considere a sub-rotina `rotx` indicada a seguir:

```
rotx PROC
    local v[8]:byte
    lea esi,v[0]
    ...
    ret
rotx ENDP
```

 A instrução `lea esi,v[0]` carrega em ESI:

- A. o primeiro elemento de `v[]`  
 B. os 4 primeiros elementos de `v[]`  
 C. o endereço do 1º elemento de `v[]`  
 D. nenhum dos valores anteriores

- [10] 4. Considere o seguinte protótipo em linguagem C/C++:

```
extern "C" int func(char texto[64], int p);
```

Indique o cabeçalho da correspondente sub-rotina em linguagem *assembly*.

- A. `func proc C texto:byte, p:sdword`
- B. `func proc C p:sdword, texto:byte`
- C. `func proc C uses edi ebx texto:ptr byte, p:sdword`
- D. `func proc stdcall uses edi esi texto:ptr byte, p:sdword`

- [10] 5. Considere a sequência de instruções:

```
mov ecx,100
@@: mov al,[esi]
inc esi
mov [edi],al
inc edi
loop @b
```

Qual dos seguintes fragmentos lhe é equivalente?

- |                          |                         |                          |                         |
|--------------------------|-------------------------|--------------------------|-------------------------|
| A. <code>cld</code>      | B. <code>cld</code>     | C. <code>cld</code>      | D. <code>std</code>     |
| <code>mov ecx,100</code> | <code>mov ecx,25</code> | <code>mov ecx,100</code> | <code>mov ecx,25</code> |
| <code>rep lodsb</code>   | <code>@@: lodsd</code>  | <code>rep movsd</code>   | <code>rep movsd</code>  |
| <code>rep stosb</code>   | <code>stosd</code>      |                          |                         |
|                          | <code>loop @b</code>    |                          |                         |

- [10] 6. Considere a execução do seguinte fragmento:

```
.data
var byte "ABCDEFGH",0
.code
mov edi, offset var
mov al, "E"
repne scasb
```

Sabendo que o endereço de `var` é 03714902H, qual é o valor final do registo EDI?

- A. 03714903H   B. 0371490CH   C. 03714906H   D. 03714907H

- [10] 7. Chama-se pilha de vírgula flutuante:

- A. ao conjunto de registos da unidade de vírgula flutuante
- B. à zona de memória externa apontada pelo registo FSP
- C. a uma zona de memória externa onde se guardam valores reais
- D. ao conjunto de registos do processador quando armazenam pelo menos um valor real

- [10] 8. Qual das seguintes instruções é válida?

- A. `fild ebx`   B. `fild 2`   C. `fstp real10 ptr [esi+ecx]`   D. `fsin ST(2)`

- [10] 9. Para um jogo de computador, qual das seguintes características de um computador é mais importante?

- A. débito   B. tempo de resposta   C. fiabilidade   D. consumo de energia

[10] 10. Considere uma sub-rotina de tratamento de imagem semelhante às estudadas nas aulas práticas:

```
afunc1 proc pixels:ptr byte, largura:dword, altura:dword
```

Qual dos seguintes fragmentos coloca o primeiro pixel de cada linha a azul?

- |  |  |
|--|--|
| <p>A.</p> <pre>mov edi, pixels mov ecx, altura mov eax, 0000000FFh B: stosd add edi, largura loop B</pre>              | <p>B.</p> <pre>mov esi, pixels mov ecx, altura mov eax, largura shl eax, 2 @@: mov dword ptr [esi],0000000FFh add esi, eax loop @B</pre> |
| <p>C.</p> <pre>mov edi, pixels mov ecx, altura mov eax, largura @@: mov byte ptr [edi],0FFh add edi, eax loop @B</pre> | <p>D.</p> <pre>mov edi, pixels mov ecx, largura mov eax, altura @@: mov byte ptr [edi*4],0FFh add edi, eax loop @B</pre>                 |

11. Considere o seguinte programa:

```
1 include mpcp.inc
2 .data
3 SEQ sdword 1, 7, -5, -1, -6, 0, 3, 1
4 LM sdword lengthof SEQ dup (?)
5 msg BYTE "%d", 13, 10, 0
6
7 .code
8 local_max proc uses ebx esi pt:ptr sdword
9 mov eax, 1
10 mov esi, pt
11 mov ebx, [esi]
12 .if (ebx <= sdword ptr [esi-4]) || (ebx <= sdword ptr [esi+4])
13 xor eax, eax
14 .endif
15 ret
16 local_max endp
17
18 main: xor ebx, ebx
19 mov edi, offset LM
20 mov esi, offset SEQ
21 add esi, type SEQ
22 mov ecx, lengthof SEQ
23 sub ecx, 2
24 cld
25 L1: invoke local_max, esi
26 add ebx, eax
27 .if eax==0
28 add esi, 4
29 .else
30 movsd
31 .endif
32 L2: loop L1
33 invoke printf, offset msg, ebx
34 invoke _getch
35 invoke ExitProcess, 0
36 end main
```

- [10] (a) O número de vezes que a sub-rotina `local_max` é executada é:  
A. 5      B. 6      C. 7      D. 8
- [10] (b) Após execução do programa a sequência LM possui os valores:  
A. 3      B. 7, -1, 3      C. 7, 3      D. 1, -5, -6, 0, 1
- [10] (c) Assuma que a sequência `SEQ` está em memória a partir do endereço `00BBC008H`. Considerando que antes da primeira invocação de `local_max` `ESI=007A8442H`, após a execução de `local_max` o valor de `ESI` é:  
A. `00BBC00CH`      B. `007A8446H`      C. `007A8442H`      D. `00BBC008H`
- [10] (d) Durante a execução da linha 25 (invocação e execução de `local_max`), o número de acessos à pilha é:  
A. 8      B. 9      C. 3      D. 10

## Parte II — Exercício de programação

**Atenção:** Responder em folha separada.

12. Considere uma sequência de pixels de uma imagem codificada em RGB. A informação de cada pixel, composta pelas componentes RGB e opacidade, é guardada em 4 bytes. R ocupa o byte com endereço mais baixo e a opacidade ocupa o byte com endereço mais alto. Pretende-se converter a imagem para a codificação CMYK. Para esta conversão, a componente C (*cyan*) é calculada por

$$C = \begin{cases} 1 - \frac{R}{\max(R,G,B)} & \text{se } \max(R,G,B) \neq 0 \\ 0 & \text{se } \max(R,G,B) = 0 \end{cases}$$

- [30] (a) Implemente a sub-rotina `RGB2C` que recebe o endereço de um pixel e calcula o valor de C.  
O protótipo desta sub-rotina é: `RGB2C proto pix:ptr byte`.
- [30] (b) Implemente a sub-rotina `fillC` que recebe o endereço de uma sequência de N pixels e preenche uma sequência, endereçada por `ptC`, com os respectivos valores da componente C utilizando a sub-rotina `RGB2C`.  
O protótipo da sub-rotina é: `fillC proto pt:ptr byte, N:dword, ptC:ptr real8`.

Fim do enunciado