

Nome: _____ Nº de estudante: _____

Atenção: Este teste tem 12 questões em 5 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. Considere a declaração `rot2016 PROC A:word, B:dword`. No código desta sub-rotina, o endereço de A é dado por:
- A. `EBP+6` B. `EBP+8` C. `EBP-8` D. `EBP+4`
- [10] 2. Considere que `ECX=8`, `ESI=7777AF80h`, `EDI=7777EEEE0h` e `flag D=0`. Indique o valor de `ESI` após a execução de `REP STOSW`.
- A. `7777AF90h` B. `7777AF88h` C. `7777AF80h` D. `7777AF86h`
- [10] 3. Cada pixel de uma imagem ocupa 4 bytes em memória. O endereço do primeiro pixel de uma linha é dado por `ESI` e o número de pixels de cada linha é dado por `ECX`. Indique qual das seguintes opções determina o endereço do último pixel dessa linha.
- A. `dec ecx` B. `mov eax, 4` C. `add esi, ecx` D. `shl ecx, 2`
 `shl ecx, 4` `mul ecx` `shl esi, 2` `add esi, ecx`
 `add esi, ecx` `add esi, eax` `add esi, -4` `sub esi, 4`
- [10] 4. Indique qual das seguintes instruções é inválida.
- A. `fiadd eax` B. `fdivr word ptr [esi]` C. `fabs` D. `fstp st(0)`
- [10] 5. Considere a declaração `frase byte "Hello World!",10,13,0` e a sequência de instruções:
- ```

mov ecx, -1
cld
mov edi, offset frase
xor eax, eax
repne scasb

```

 Quantas vezes é executada a instrução `scasb`?

- A. 14    B. 16    C. 15    D. 12

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

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

Relembrando: Cada ponto da imagem (pixel) é representado por quatro bytes consecutivos, com o seguinte significado: 1. valor da componente B (azul), 2. valor da componente G (verde), 3. valor da componente R (vermelho) e 4. valor da transparência.

Qual dos seguintes fragmentos coloca o primeiro pixel da imagem a verde?

- |                                                                                                                                                                             |                                                                                                                                                                          |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A. <code>mov edi, pixels</code><br/> <code>mov dword ptr [edi], 0000FF00h</code></p> <p>C. <code>mov edi, pixels</code><br/> <code>mov word ptr [edi], 0FF00h</code></p> | <p>B. <code>mov edi, pixels</code><br/> <code>mov byte ptr [edi+1], 0FFh</code></p> <p>D. <code>mov edi, pixels</code><br/> <code>mov word ptr [edi+2], 00FFh</code></p> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- [10] 7. Considere o fragmento de código seguinte:

|                                                                       |                                                                           |
|-----------------------------------------------------------------------|---------------------------------------------------------------------------|
| <pre>.data string1 byte "aaa bbb ccc",0  .code cld xor ecx, ecx</pre> | <pre>mov esi, offset string1 .repeat   lodsb   inc ecx .until al==0</pre> |
|-----------------------------------------------------------------------|---------------------------------------------------------------------------|

Qual é o valor de ECX após a execução das instruções?

- A. 00000000Bh    **B. 00000000Ch**    C. 000000009h    D. 000000002h

- [10] 8. Sabendo que A, B, X, E e RES são do tipo `real8` e D do tipo `sdword`, indique qual das seguintes sequências de instruções calcula

$$RES = \frac{(B - A) \times X}{(D + A) + E}$$

- |                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A. <code>fld A</code><br/> <code>fld B</code><br/> <code>fsub</code><br/> <code>fmul X</code><br/> <code>fld D</code><br/> <code>fld A</code><br/> <code>fadd</code><br/> <code>fadd E</code><br/> <code>fdiv</code><br/> <code>fstp RES</code></p> | <p>B. <code>fld B</code><br/> <code>fld A</code><br/> <code>fsub</code><br/> <code>fmul X</code><br/> <code>fld D</code><br/> <code>fld A</code><br/> <code>fadd</code><br/> <code>fadd E</code><br/> <code>fdiv</code><br/> <code>fstp RES</code></p> | <p>C. <code>fld B</code><br/> <code>fld A</code><br/> <code>fsub</code><br/> <code>fmul X</code><br/> <code>fild D</code><br/> <code>fld A</code><br/> <code>fadd</code><br/> <code>fadd E</code><br/> <code>fdiv</code><br/> <code>fstp RES</code></p> | <p>D. <code>fld B</code><br/> <code>fld A</code><br/> <code>fsub</code><br/> <code>fmul X</code><br/> <code>fild D</code><br/> <code>fild D</code><br/> <code>fadd</code><br/> <code>fadd E</code><br/> <code>fdivr</code><br/> <code>fstp RES</code></p> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- [10] 9. A que é equivalente a instrução `fdiv`?

- |                                                                                       |                                                                                       |
|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <p>A. <code>fdiv st(0),st(1)</code><br/> <b>C. <code>fdivp st(1),st(0)</code></b></p> | <p>B. <code>fdiv st(1),st(0)</code><br/> <b>D. <code>fdivp st(0),st(1)</code></b></p> |
|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|

- [10] 10. Quais os parâmetros que mais influenciam o tempo necessário para aceder à informação de um setor aleatório num disco magnético?

**A. O tempo médio de busca e a velocidade de rotação.**

B. A velocidade de rotação e a taxa de transferência.

C. A taxa de transferência e a latência do controlador.

D. A latência do controlador e o tempo médio de busca.

11. Considere o seguinte programa:

```
1 include mpcp.inc
2 .data
3 seq SDWORD -1, -2, 0, 4, 2, 10, 8
4 fmt BYTE "%d ", 0
5 CRLF BYTE 13,10,0
6 funcx PROC seqptr: ptr sdword, NElem: dword
7 .code
8 main: invoke funcx, offset seq, lengthof seq
9 mov esi, offset seq
10 mov ecx, eax
11 @@: push ecx
12 invoke printf, offset fmt, sdword ptr [esi]
13 add esi, 4
14 pop ecx
15 loop @B
16 invoke printf, offset CRLF
17 invoke _getch
18 invoke ExitProcess, 0
19
20 funcx PROC uses edi esi seqptr: ptr sdword, NElem: DWORD
21 mov eax, 1
22 mov ecx, NElem
23 dec ecx
24 mov edi, seqptr
25 mov esi, edi
26 add esi, 4
27 mov edx, [edi]
28 .WHILE (ecx > 0)
29 cmp edx, [esi]
30 jge @F
31 inc eax
32 mov edx, [esi]
33 add edi, 4
34 mov [edi], edx
35 @@: add esi, 4
36 dec ecx
37 .ENDW
38 ret
39 funcx endp
40 end main
```

- [10] (a) Sabendo que `offset seq` é 20003A00h, qual é o valor de `offset CRLF`?  
A. impossível de determinar   B. 20003A08H   C. 20003A1FH   D. 20003A20H
- [10] (b) Quantas vezes é executado o corpo do ciclo `WHILE` da sub-rotina `funcx`?  
A. 7   B. 6   C. 4   D. 3
- [10] (c) Quantos bytes ocupa a moldura da execução de `funcx`?  
A. 8   B. 16   C. 24   D. 4
- [10] (d) Que sequência de valores é apresentada no monitor?  
A. -1 0 4 10   B. -1 -2 0 4 2 10 8  
C. 4 2 10 8   D. -2 2 4 8

## Parte II — Exercício de programação

**Atenção:** Responder em folha separada.

12. Um conjunto de  $n$  pontos  $(x_i, y_i)$  no plano é representado por uma sequência de números reais organizados pela seguinte ordem:

$$(x_1, y_1, x_2, y_2, x_3, y_3, \dots, x_n, y_n)$$

- [30] (a) Escrever uma sub-rotina que determina a menor coordenada  $x_i$  de um grupo de pontos representado conforme indicado acima. A sub-rotina tem o seguinte protótipo:

menorX PROTO seqpt:ptr real4, npt:DWORD

**Solução:**

```

1 menorX PROC seqpt: ptr real4, npt: DWORD
2 mov ecx, npt
3 mov edx, seqpt
4 fld real4 ptr [edx] ; primeiro valor
5 dec ecx
6 add edx, 8 ; outros valores a 8 bytes de distância
7 .WHILE ecx > 0
8 fld real4 ptr [edx]
9 fcomi ST(0), ST(1) ; afeta "flags" do CPU
10 jae @F
11 fxch
12 @@: fstp st(0) ; e é eliminado
13 add edx, 8
14 dec ecx
15 .ENDW
16 ret
17 menorX endp

```

- [30] (b) Assumir que já existem as sub-rotinas:

- maiorX PROTO seqpt:ptr real4, npt:DWORD
- menorY PROTO seqpt:ptr real4, npt:DWORD
- maiorY PROTO seqpt:ptr real4, npt:DWORD

Para um conjunto de `npt` pontos endereçados por `seqpt`, estas sub-rotinas determinam respetivamente a maior coordenada  $x_j$ , a menor coordenada  $y_k$  e a maior coordenada  $y_m$ .

Escrever um programa que determina e apresenta no monitor o comprimento (diferença  $x_j - x_i$ ) e a largura (diferença  $y_m - y_k$ ) do menor retângulo que inclui esse conjunto de pontos. O início do programa deve ser o seguinte:

```

include mpcp.inc
.data
pontos REAL4 0.0, 0.0, -2.1, 4.0, 2.2, 2.7, 1.5, -3.2, -1.5, -0.9
...

```

**Solução:**

```

1 include mpcp.inc
2 menorX PROTO seqpt:ptr real4, npt:DWORD
3 maiorX PROTO seqpt:ptr real4, npt:DWORD

```

```
4 menorY PROTO seqpt:ptr real4, npt:DWORD
5 maiorY PROTO seqpt:ptr real4, npt:DWORD
6 .data
7 pontos REAL4 0.0, 0.0, -2.1, 4.0, 2.2, 2.7, 1.5, -3.2, -1.5, -0.9
8 X_min real4 ?
9 Y_min real4 ?
10 comprimento real8 ?
11 largura real8 ?
12 fmt byte "Comprimento=%f Largura=%f", 13, 10, 0
13 .code
14 main:
15 invoke menorX, offset pontos, lengthof pontos / 2
16 fstp X_min ; esvaziar pilha UVF
17 invoke maiorX, offset pontos, lengthof pontos / 2
18 fsub X_min
19 fstp comprimento ; esvaziar pilha UVF
20
21 invoke menorY, offset pontos, lengthof pontos / 2
22 fstp Y_min ; esvaziar pilha UVF
23 invoke maiorY, offset pontos, lengthof pontos / 2
24 fsub Y_min
25 fstp largura ; esvaziar pilha UVF
26
27 invoke printf, offset fmt, comprimento, largura
28 invoke _getch
29 invoke ExitProcess,0
30 end main
```

FIM.