

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. Quantos bytes de memória são reservados pela declaração `var SDWORD 3 dup(2)?`

A. 3 B. 8 C. 6 D. 12

[10] 2. Qual das seguintes instruções dá erro ao compilar?

 A. `mov [esi],[edi]` B. `mov ax,[esi]` C. `mov [edi],al` D. `mov ax,8888H`

 [10] 3. Qual o valor do registo `al` após execução do seguinte código:

```
mov al, 4AH
ror al, 2
adc al, 0AH
```

A. 2FH B. 9CH C. 9DH D. 34H

[10] 4. Para saber se o resultado de uma operação entre operandos com sinal dá zero, basta verificar se:

A. ZF=1 B. ZF=1 e SF=0 C. ZF=0 D. ZF=0 e SF=0

 [10] 5. Qual o valor de `eax` após a execução do seguinte código:

```
mov eax,v1
cmp eax,v2
cmovng eax,v2
```

 A. `min(v1, v2)` B. `v1` C. `max(v1, v2)` D. `v2`

 [10] 6. Quais os valores de `eax` e `edx` após a execução do seguinte código:

```
mov eax,-1
mov edx,100
imul edx
```

A. 100 e 0 B. -100 e -1 C. -100 e 0 D. 100 e -1

[10] 7. Qual o valor de `edx` após a execução do seguinte código:

```
mov  eax, -101
cdq
inc  eax
inc  edx
```

- A. -1 B. -102 C. -100 **D. 0**

[10] 8. Considere o fragmento de código:

```
xor  edx, edx
mov  ebx, 7
.if  eax>ebx
    inc  edx
.endif
```

Após execução do código `edx` é diferente de zero se `eax` tiver o valor:

- A. -7** B. 0 C. 6 D. 7

[10] 9. Indique o conteúdo de `eax` após execução do seguinte fragmento de código:

```
.data
val  word 2
.code
mov  eax, 0F000FFFFh
mul  val
```

- A. 00001FFFEh B. 0F000FFFEh C. 0000FFFF0h D. 0E001FFFEh

[10] 10. Indique qual dos fragmentos de código coloca em `eax` o menor valor do tipo `SDWORD`.

- | | | | |
|------------------------------|------------------------------------|---------------------------|-----------------------------------|
| A. <code>xor eax, eax</code> | B. <code>mov eax, 7fffffffh</code> | C. <code>stc</code> | D. <code>mov eax, 1</code> |
| <code>stc</code> | <code>neg eax</code> | <code>sub eax, eax</code> | <code>ror eax, 1</code> |
| <code>ror eax, 1</code> | | <code>rcr eax, 1</code> | |

11. Considere o seguinte programa:

```
1  include mpcp.inc
2      .data
3  msg BYTE "jh78-34h-f49-g78X8-gLG-HYrtE"
4  fmt BYTE "%d",13,10,0
5
6      .code
7  main:  mov     edi, offset msg
8         mov     ecx, lengthof msg
9         xor     eax, eax
10 L1:    mov     dl, [edi]
11        cmp     dl, 'a'      ; código ASCII de carácter a
12        jb      @F
13        cmp     dl, 'z'      ; código ASCII de carácter z
14        ja      @F
15        inc     eax
16 @@:    inc     edi
17        loop    L1
18        invoke  printf, offset fmt, eax
19        invoke  ExitProcess, 0
20 end main
```

- [10] (a) Quantas vezes é executada a instrução `mov dl,[edi]` ?
A. 3 **B. 28** C. 30 D. 33
- [10] (b) Qual é o tamanho da zona de dados ocupada por este programa?
A. 30 bytes B. 37 bytes **C. 33 bytes** D. impossível determinar
- [10] (c) A vista de `disassembly` mostra que a invocação `invoke printf,offset fmt,eax` dá origem a:
- | | | | |
|----------|----------------|------|-----------------------|
| 00DD101C | 50 | push | eax |
| 00DD101D | 68 1C 40 DD 00 | push | offset fmt (0DD401Ch) |
| 00DD1022 | E8 1D 00 00 00 | call | printf (0DD1044h) |
| 00DD1027 | 83 C4 08 | add | esp,8 |
- Qual é o tamanho do código-máquina necessário para invocar a função `printf`?
A. 4 bytes B. 16 bytes C. 28 bytes **D. 14 bytes**
- [10] (d) Qual é o resultado apresentado no monitor no final da execução?
A. 8 B. 5 C. 14 D. 9

Parte II — Exercício de programação

Atenção: Responder no enunciado.

- [60] 12. Escrever um programa que para um dado valor positivo `limiar`, coloca todos os elementos de uma sequência de valores do tipo `SDWORD` que fiquem fora do intervalo `[-limiar;+limiar]` no valor do extremo do intervalo mais próximo do número original. O programa deve indicar quantos elementos são alterados e dar tempo ao utilizador para ler a mensagem.

Exemplo: Para a sequência indicada abaixo, o programa deve imprimir **Alterados 4 elementos**.

```
include mpcp.inc
.data
seq      SDWORD  15, -19, 8, 71, -100, 20, 41, -11, -20, -54
limiar   SDWORD  20      ; sempre positivo
formato  BYTE   "Alterados %d elementos.", 13, 10, 0

.code
main:
    mov edi, offset seq
    mov ecx, lengthof seq
    mov eax, limiar
    xor ebx, ebx
@@: .IF SDWORD PTR [edi] > eax
        mov     [edi], eax
        inc ebx
    .ENDIF
    neg eax
    .IF SDWORD PTR [EDI] < eax
        mov [edi], eax
        inc ebx
    .ENDIF
    neg eax
    add edi, 4
    loop @B

;;    imprimir resultado
    invoke printf, offset formato, ebx
    invoke _getch
    invoke ExitProcess,0

end main
```