

Nome:

Nº:

**Duração Total:** 00:40 m**Sem consulta****Teste 2**

1. A janela abaixo representa o segmento de dados de um programa:

```
dseg      segment para public 'data'
          var1          db      128
          var2          dw      256
          var3          dd      10101010b
          var4          sbyte   -32
          var5          sword   -32
          var6          dword   64
          var7          db      3 dup (2 dup (10h))
          var8          REAL4   -9.0
dseg      ends
```

Tendo em consideração a representação deste segmento de dados, no segmento com endereço 3030 h, indique:

a. Qual o endereço de deslocamento relativo a var1?

- i. 0000h
- ii. 3030h:0001h
- iii. 0001h
- iv. Nenhum dos anteriores

b. O conteúdo em memória do endereço 3030h:0001h?

- |         |         |                            |
|---------|---------|----------------------------|
| i. 01 h | ii. 00h | iii. Nenhum dos anteriores |
|---------|---------|----------------------------|

c. O conteúdo em memória do endereço 3030h:0003h?

- |          |        |                           |
|----------|--------|---------------------------|
| iv. AA h | v. 00h | vi. Nenhum dos anteriores |
|----------|--------|---------------------------|

d. O conteúdo em memória do endereço 3030h:0007h?

- |           |           |                           |
|-----------|-----------|---------------------------|
| vii. 20 h | viii. E0h | ix. Nenhum dos anteriores |
|-----------|-----------|---------------------------|

e. A representação de var5 em memória?

Endereço	Conteúdo	Endereço	Conteúdo
a. 0008	E0	b. 0008	E0
0009	00	0009	FF

f. Qual o número de bits ocupados pela variável real 4?

- i. 4
- ii. 16
- iii. 32
- iv. Nenhum dos anteriores

g. Qual o conteúdo do endereço 3031:0001h?

- i. 10 h
- ii. 80 h
- iii. 00 h
- iv. Nenhum dos anteriores

2. Tendo em consideração o programa abaixo, responda às seguintes questões:

```
1: DATA_HERE      SEGMENT
2:                  v1    dw 8 dup (?)
3: DATA_HERE      ENDS
4: STACK_HERE      SEGMENT STACK
5:                  DW 40 DUP(0)
6: STACK_HERE      ENDS
7: CODE_HERE       SEGMENT
8:                  ASSUME CS:CODE_HERE, DS:DATA_HERE, SS: STACK_HERE
9: Main            Proc
10:                 mov ax, DATA_HERE
11:                 mov ds, ax
12:
13:                 xor si, si
14:                 mov ax, 8
15:                 mov bx, 2
16:                 mov cx, 3
17:                 mov dx, 3
18:         ciclo:   call calculo
19:
20:                 sub  ax, bx
21:                 mul  dl
22:
23:                 mov  v1[si], ax
24:                 add  si, 2
25:                 mov  v1[si], bx
26:                 add  si, 2
27:                 mov  v1[si], cx
28:                 add  si, 2
29:                 mov  v1[si], dx
30:                 add  si, 2
31:                 loop ciclo
32:
33:                 mov  ah, 4ch
34:                 int  21h
```

```

35:Main          endp
36:
37: calculo      PROC
38:             pushf
39:             push cx
40:             push bx
41:
42:             mov dx,0
43:             add ax, cx
44:             div bx
45:             add cx,1
46:
47:             pop bx
48:             pop cx
49:             popf
50:             ret
51: calculo      ENDP
52:CODE_HERE     ENDS
53:             END Main

```

- a. Indique o número de vezes que “ciclo” é realizado
  - i. 8
  - ii. 2
  - iii. 3
  - iv. Nenhum dos anteriores
  
- b. A instrução “ret” (linha 50) é facultativa
  - i. Sim
  - ii. Não
  
- c. É possível substituir as 4 instruções “add si,2” por uma “add si, 8” na linha anterior ao loop
  - i. Sim
  - ii. Não
  
- d. Indique o valor dos registos abaixo na 1ª iteração do ciclo
  - i. AX (linha 23) \_\_\_\_\_
  - ii. BX (linha 25) \_\_\_\_\_
  - iii. CX (linha 27) \_\_\_\_\_
  - iv. DX (linha 29) \_\_\_\_\_
  
- e. Indique o valor dos registos abaixo na 2ª iteração do ciclo
  - i. AX (linha 23) \_\_\_\_\_
  - ii. BX (linha 25) \_\_\_\_\_
  - iii. CX (linha 27) \_\_\_\_\_
  - iv. DX (linha 29) \_\_\_\_\_

BOA SORTE! ☺