

Nome completo:

Nº Aluno:

Sala:

Deve assinalar a resposta correcta, circundando-a. Se precisar de alterar alguma resposta deve riscá-la e circundar a nova resposta. As respostas incorrectas, serão penalizadas.

Duração Total Exame: 1h:30 m

Sem consulta

23 de Maio de 2015

1. Diga qual o número máximo de bytes que consegue endereçar no 8086.

a. 8 Bytes	b. 16 Bytes	c. 256 Bytes
d. 64 KBytes	e. 1 MegaBytes	f. Nenhuma das opções

2. Das seguintes instruções diga quais são as possíveis.

a. mov BX, [AX]	b. mov BL, DS:[AX]	c. mov AL, DS:[DX]
d. mov BX, [DL]	e. mov BL, DS:[BP]	f. mov CX, [SI]

3. Sabendo que pretende somar o valor presente na variável var1 (byte) ao valor na variável var2(byte), indique qual(is) a(s) opção(ões) possível(is).

a. add var1, var2	b. mov AL, var1 add BX, var2 add BX, AL	c. mov AL, var1 mov BL, var2 add AL, BL adc AL, 0
d. mov AL, var1 mov BL, var2 xor AH, AH xor BH, BH add BX, AX	e. mov AH, 0 mov AL, var1 add AL, var2 add AH, 0	f. xor DH, DH mov DL, var1 add DL, var2 adc DH, 0

4. Sabendo que pretende multiplicar 350 por 2 indique qual(is) a(s) opção(ões) possível(is).

a. mov AX, 350 mov BX, 2 mul BX	b. mov AL, 350 mul AL, 2	c. mov BX, 350 mul BX, 2
d. mul 350, 2	e. mov AX, 350 mul 2	f. mov AX, 350 mov bl, 2 mul bl

5. Sabendo que pretende dividir 751 por 2 indique qual(is) a(s) opção(ões) possível(is).

a. mov AX, 751 mov BL, 2 div BL	b. mov DX, 0 mov AX, 751 mov BX, 2 div BX	c. mov DX, 0 mov AX, 751 mov CX, 2 div CL
d. mov DX, 0 mov AX, 751 div 2	e. mov AL, 751 mov DL, 2 div DL	f. mov AX, 751 div 2

6. Tendo em consideração o programa abaixo

```
.8086
.model small
.stack 2048

dados segment para 'data'
    Address1 dw 0FFFFh, 15 dup(1)
    Address2 dw 6, 15 dup(1)
    Address3 dw 9 dup (?)
dados ends

codigo segment para 'code'
main proc
    assume cs:codigo, ds:dados
    mov ax,dados
    mov ds,ax

    mov cx,16
    xor si,si
    xor dx,dx
ciclo: mov ax,dx
    xor dx,dx
    add ax, Address1[si]
    adc dx,0
    add ax, Address2[si]
    adc dx,0
    mov Address3[si], ax
    add si,02h
    sub cx,1
    loop ciclo

    mov ah,4Ch
    int 21h
main endp
codigo ends
end main
```

I. Identifique o tipo da variável Address1

a. Word	b. Byte
c. Vetor de words	d. Vetor de byte

II. Indique o número de elementos de Address1

a. 1	b. 65535
c. 15	d. 16

III. Qual o número de vezes que o *loop ciclo* é executado?

a. 16	b. O valor existente em SI
c. 8	d. 1

IV. Qual o conteúdo de AX passado para a 1ª posição de Address3?

a. 0002h	b. 001Eh
c. 0005h	d. 0030h

V. Qual o conteúdo de DX no final da primeira iteração do ciclo?

a. 0000h	b. 0001h
c. O valor de AX	d. Nenhuma das opções

VI. É possível substituir “add ax, Address1[si]”, por “add al, Address1[si]”?

a. Sim	b. Não
--------	--------

VII. É possível substituir em todo o código o registo si por bx”?

a. Sim	b. Não
--------	--------

7. Considere o seguinte segmento de dados de um programa:

```
dseg      segment para public 'data'
                var1      dw          40h
                var2      db          0AAh
                var3      sword      E0h
                var4      byte      0,9,16
                var5      real4      4.0
                var6      word      2 dup (0CCh)
dseg      ends
```

Supondo que o DS deste programa assume o valor 0ABC, indique:

I. O conteúdo em memória do endereço 0ABCh:0001h?

a. 40h	b. 00h	c. AAh
--------	--------	--------

II. A representação de var4 em memória em hexadecimal?

Endereço	Conteúdo	Endereço	Conteúdo
a. 0005	00	b. 0005	00
0006	09	0006	09
0007	16	0007	10

Formato geral:

• **ADD DEST, ORIG**

Flags alteradas: SF, ZF, PF, AF, CF, OF.

• **ADC DEST, ORIG**

Flags alteradas: SF, ZF, PF, AF, CF, OF.

• **SUB DEST, ORIG**

Flags alteradas: SF, ZF, PF, AF, CF, OF.

• **(I)MUL OPER**

Ação:

DEST:= DEST + ORIG

DEST:= DEST + ORIG + Flag Carry

DEST:= DEST – ORIG

Se OPER for de 8 bits AX:= AL *OPER

Se OPER for de 16 bits DX:AX:=AX*OPER

Flags alteradas: As flags CF, OF, SF, ZF, PF e AF ficam com valores indefinidos.

• **(I)DIV OPER**

Se OPER for de 8 bits divide AX por OPER, ficando o quociente em AL e o resto em AH.

Se OPER for de 16 bits divide DX:AX por OPER, ficando o quociente em AX e o resto em DX.

Flags alteradas: CF, SF, ZF, PF, AF, OF.

• **LOOP label**

De cada vez que a instrução LOOP é executada o conteúdo do registo CX é decrementado um valor.

- Se depois de decrementado o conteúdo do registo CX o seu valor for diferente de zero, a instrução LOOP implementa um jump para o endereço referenciado por <label>.

Não afecta as flags.