



Nome: _____

Data: ____ \ ____ \ ____

Questão 1) Converta para a base 10 os inteiros em representação complemento a dois.

- a) 100101 c) 111111
b) 010011

a) -27 b) 19 c) -1

Questão 2) Converta para a representação em binário de ponto fixo (use 4 bits para a parte decimal e 4 bits para a parte inteira).

- a) 9,25 c) 11,125
b) 0,375

a) 10010100 b) 00000110 c) 10110010

Questão 3) Encontre uma aproximação para os seguintes valores usando 1 bit para o sinal, 4 bits para o expoente (polarizado em 7) e 8 bits para a mantissa (normalizada).

- a) 13,25 c) 0,625
b) 1,875

a) 0101010101000 b) 0011111100000 c) 0011001000000

Questão 4) Escreva um programa para computar a expressão

$$X = (A + C)/(D \cdot E)$$

nos seguintes formatos de Assembly

- a) Stack machine
push D
push E
mul
push A
push B
add
div
pop X

b) Acumulador
load D
mul E ; D*E
store X ; X = D*E
load A
add C ; ACC = A + C

div X ; ACC = (A+C)/(D*E)

store X
c) Register-memory
load R1, A
add R1, B
load R2, D
mul R2, E
div R1, R2
store X, R1
d) Load-Store
load R1, A
load R2, B
add R3, R1, R2
load R1, D
load R2, E
mul R4, R1, R2
div R3, R3, R4
store X

Questão 5) Transcreva os trechos de códigos (em C) para Assembly x86:

a)

```
if(a > 0)
    a = a + 10*b;
else
    a = a * (10+b);

cmp ecx, 0
JLE else
mov ecx, a
mov eax, b
mul 10 ; eax = 10*b
add ecx, eax ; ecx = a + 10*b
mov a, ecx
jmp endif
else:
    mov eax, a
    mov ebx, b
    add ebx, 10 ; ebx = b + 10
    mul ebx ; eax = a*(b + 10)
    mov a, eax
endif.
```

b)

```
for(i=0; i < 10; i++) {
    a = a + 3;
```

```

        b = b * 2;
    }

    mov ecx, a
    mov ebx, 0
    mov eax, b
    LOOP_FOR:
    add ecx, 3 ; ecx = a + 3
    mul 2 ; eax = b*2
    inc ebx ;
    cmp ebx, 10
    JL LOOP_FOR
    mov a, ecx
    mov b, eax
    c)

for(i=100; i >= 0; i-= 10) {
    v[i] = v[i-1]+a;
}

mov esi, 100
mov eax, a
LOOP_FOR:
mov edi, esi
dec edi
mov ebx, [v+edi]
add [v+esi], ebx
add [v+esi], eax
sub esi, 10 ;
cmp esi, 0
JGE LOOP_FOR

```

Questão 6) Crie um procedimento para computar o tamanho de uma string Assembly x86.

strlen:

```

mov esi, 0 ; counter
mov ebx, [esp+4]
WHILE:
    mov ecx, [ebx+esi]
    cmp ecx, 0
    je END
    inc esi
    jmp WHILE
END:
    mov eax, esi ; strlen
    ret

```

Questão 7) Crie um procedimento usando a FPU para computar o volume da pirâmide quadrangular

$$V = \frac{l^2 h}{3}$$

(h é a altura e l é o lado da base).

volume_piramide:

```

fld qword[esp+4] ; l
fmul ST0 ; l*l
fld qword[esp+12] ; h, l*l
fmulp ST1 ; h*l*l
fld qword [three] ; 3.0, h*l*l
fdivp ST1 ; h*l*l/3.0
fst qword [V] ; V = h*l*l/3.0

```