

**Disciplina:** ELE1717 - Sistemas Digitais  
**Aluno:**

**Período:** 2018.1  
**Data:** 27/03/2018

**1** (0,5) - Desenvolva um código fonte em Assembly para retornar o valor e a posição do maior número em uma lista de 20 valores possíveis (em caso de números iguais retorne a posição de qualquer um deles). O valor do número deverá ser colocado no endereço 232 da memória e a sua posição na lista (entre 1 e 20) no endereço 233 da memória.

**2** (0,5) - Desenvolva um código fonte em Assembly para verificar se uma palavra (de no máximo 20 caracteres) é ou não palíndromo. Caso a palavra seja palíndromo, escreva no endereço 232 da memória o valor 48, do contrário escreva no endereço 232 da memória o valor 49.

**3** (0,5) - Desenvolva um código fonte em Assembly para multiplicar dois números de 8bits e obter um resultado em um número de 16bits. Os valores de 8bits devem ser colocados nos endereços 0 e 1 da memória e o resultado de 16bits deve ser colocado nos endereços 232(MSB) e 233(LSB).

**4** (0,5) - Desenvolva um código fonte em Assembly para calcular o número de Fibonacci (que resulte em um valor máximo de 16bits). O valor de entrada para a sequência de Fibonacci deverá ser colocado no endereço 0 da memória. O resultado de 16bits deve ser colocado nos endereços 232(MSB) e 233(LSB).

$$F(n) \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ F(n-1) + F(n-2), & n > 1 \end{cases} \quad (1)$$

### Observações

- A interpretação das questões faz parte da prova;

Id	Instrução	Descrição
1	MOV reg, regX	reg=regX
2	MOV reg, [regX]	reg=mem[regX]
3	MOV reg, [address]	reg=mem[address]
4	MOV reg, constant	reg=constant
5	MOV [reg], regX	mem[reg]=regX
6	MOV [address], reg	mem[address]=reg
7	MOV [address], Constant	mem[address]=constant
8	ADD reg, regX	reg=reg+regX
9	ADD reg, [regX]	reg=reg+mem[regX]
10	ADD reg, [address]	reg=reg+mem[address]
11	ADD reg, constant	reg=reg+constant
12	SUB reg, regX	reg=reg-regX
13	SUB reg, [regX]	reg=reg-mem[regX]
14	SUB reg, [address]	reg=reg-mem[address]
15	SUB reg, constant	reg=reg-constant
16	MUL reg	[A]=[A]*reg
17	MUL [reg]	[A]=[A]*mem[reg]
18	MUL address	[A]=[A]*mem[address]
19	MUL constant	[A]=[A]*constant
20	AND reg, regX	reg=reg AND regX
21	AND reg, [regX]	reg=reg AND mem[regX]
22	AND reg, [address]	reg=reg AND mem[address]
23	AND reg, constant	reg=reg AND constant
24	OR reg, regX	reg=reg OR regX
25	OR reg, [regX]	reg=reg OR mem[regX]
26	OR reg, [address]	reg=reg OR mem[address]
27	OR reg, constant	reg=reg OR constant
28	XOR reg, regX	reg=reg XOR regX
29	XOR reg, [regX]	reg=reg XOR mem[regX]
30	XOR reg, [address]	reg=reg XOR mem[address]
31	XOR reg, constant	reg=reg XOR constant
32	SHL reg, regX	reg=reg<<regX
33	SHL reg, [regX]	reg=reg<<mem[regX]
34	SHL reg, [address]	reg=reg<<mem[address]
35	SHL reg, constant	reg=reg<<constant
36	SHR reg, regX	reg=reg>>regX
37	SHR reg, [regX]	reg=reg>>mem[regX]
38	SHR reg, [address]	reg=reg>>mem[address]
39	SHR reg, constant	reg=reg>>constant
40	CMP reg, regX	if reg==regX, Z=1
41	CMP reg, [regX]	if reg==mem[regX], Z=1
42	CMP reg, [address]	if reg==mem[address], Z=1
43	CMP reg, constant	if reg==constant, Z=1
44	INC reg	reg=reg+1
45	DEC reg	reg=reg-1
46	NOT reg	reg=not(reg)
47	CALL address	[SP]=PC; SP=SP-1; PC=mem[address]
48	PUSH reg	[SP]=reg; SP=SP-1
49	PUSH [reg]	[SP]=mem[reg]; SP=SP-1
50	PUSH address	[SP]=mem[address]; SP=SP-1
51	PUSH constant	[SP]=constant; SP=SP-1
52	POP reg	reg=[SP]; SP=SP+1
53	JMP address	PC=mem[address]
54	JC	PC=mem[address], if carry=true
55	JNC	PC=mem[address], if carry=false
56	JZ	PC=mem[address], if zero=true
57	JNZ	PC=mem[address], if zero=false
58	JA	PC=mem[address], if carry=false e zero=false
59	JNBE	PC=mem[address], if carry=false e zero=false
60	JAE	PC=mem[address], if carry=false
61	JNB	PC=mem[address], if carry=false
62	JB	PC=mem[address], if carry=true
63	JNAE	PC=mem[address], if carry=true
64	JBE	PC=mem[address], if carry=true ou zero=true
65	JNA	PC=mem[address], if carry=true ou zero=true
66	JE	PC=mem[address], if zero=true
67	JNE	PC=mem[address], if zero=false
68	RET	PC=[SP]; SP=SP+1
69	HLT	parar o processador

Table 1: Conjunto de instruções do processador.