Alexandre – 43551

9,7

1- // PASSOS PARA A EXECUÇÃO DE INSTRUÇÕES

```
PC = EnderecoInicial // contador de programa (RIP)
ByteDeExecução = True
while (ByteDeExecução):
    R = Memoria [PC]
    PC = PC + 1
    DeterminaTipoInstrucao (IR, TipoDaInstrução)
    AchaDado (TipoDaInstrução, IR, LocalizaçãoDado, DadoRequerido)
    if (DadoRequerido):
        dado = Memoria [LocalizaçãoDado]
    Execute (TipoDeInstrução, dado, memória, AC, PC, BitDeInstrução)
```

C !?

2- 0705_{16}

3- a) EF b) CDEF c) 89ABCDEF
d) 0123456789ABCDEF

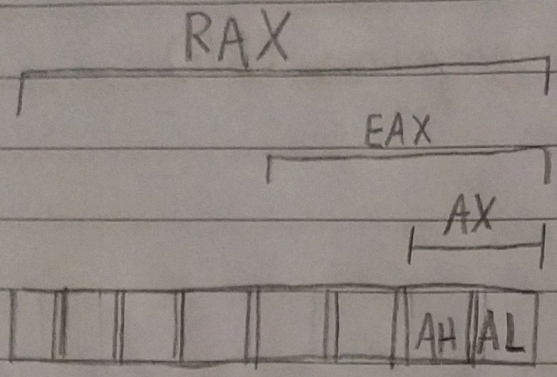
C

4- No máximo, cada registrador alcança de 1 byte ou 8 bits a 8 bytes ou 64 bits por endereço. Depende do tipo da variável ou registro.

5- a) 0 b) 0

C 1,0

5-



C 1,0

6- a) 8 b) 64 c) 16 d) 32

C 1,0

8- a) MOV R, Var carrega o endereço de Var em R ou o conteúdo de Var no caso de MOV R, [Var]

LEA - Carrega o endereço efetivo + deslocamento, se o deslocamento for 0, não é necessário adicionar. Assim, LEA R, [Var] faz o mesmo que MOV R, Var. Contudo, a instrução MOV R, Var + deslocamento não é aceita. Para isso temos que usar LEA R, [Var + deslocamento]

C 1,0

b) section .text: seção de código, onde o programa é feito
section .data: seção onde se declaram as variáveis inicializadas
section .bss: seção de variáveis não inicializadas

c) RAX: registrador conhecido como acumulador, geralmente os resultados vão para RAX
RBX: registrador que pode ser usado índice de base
RCX: registrador geralmente usado para contagem.
RDX: registrador que serve de auxiliar em algumas operações de multiplicação/divisão
RIP: registrador contador de programa, guarda o endereço da linha de instrução
RSI: registrador de índice e origem
RDI: registrador de índice e destino
RSP: registrador que aponta para o topo da pilha de memória
RBP: registrador de base da pilha
OF: flag overflow flag
DF: flag de manipulação de dados
IF: flag de interrupção
TF: flag de busca (trap)
ZF: flag pra resultado zero
PF: flag de resultado par
CF: flag de vai um

d) RIP

9) a) 0000 0000 0111 1011$_2$ = 00753$_8$ = 007B$_{16}$
b) 0000 0000 1011 1111$_2$ = 191$_{10}$ = 00BF$_{16}$
c) 137$_8$ = 95$_{10}$ = 005F$_{16}$
d) 1010 1111 1010 1101$_2$ = 127655$_8$ = -20563$_{10}$

10- a) 110 111 100 000

b) 001 000 000 010 001

c) 100 111 110 100

d) 100 111 111 C 1,0