

Nome: Cícero Higor Gomes de Sousa  
Matrícula: 21550190

### LISTA DE OC - ARM ASSEMBLY

- Eu considerei que o resultado de cada questão é armazenado em LR.

#### Questão 01

```
AREA_Variaveis
var1          DCD      100 ; valor de R1
var2          DCD      500 ;valor de R2
              ADR      R1,var1
              LDR      R1,[R1]
              ADR      R2,var2
              LDR      R2,[R2]

AREA_Instrucoes
              CMP      R1,R2 ; verifica se R1 > R2
              BLE      menor ; se R1 nao for maior,entao R2
e maior desvia o fluxo
              MOV      LR,R1 ; mostra em LR o maior
              B        FIM ; fim da condicao
menor         MOV      LR,R2 ; mostra em LR o maior
FIM
```

#### Questão 02

```
AREA_Variaveis
soma          DCD      0 ; variavel que guardara o resultado
da soma de 0 ate 100
              ADR      R0,soma
              LDR      R0,[R0]

AREA_Instrucoes

LOOP
              CMP      R0,#100 ; verifica se R0 > 100
              BGE      DONE ; desvio da condicao
              ADD      R0,R0,#1 ; soma+=1
              B        LOOP ; volta para a condicao
DONE          MOV      LR,R0 ; guarda em LR o resultado da
soma
```

### Questão 03

#### AREA\_Variaveis

```
dividendo      DCD      6 ; valor que sera verifica se eh par
ou impar
divisor        DCD      2 ; constante nao modifica, pois
queremos dividir o numero por 2
quociente      DCD      0
               ADR      R1,dividendo
               LDR      R1,[R1]
               ADR      R2,divisor
               LDR      R2,[R2]
               ADR      R3,quociente
               LDR      R3,[R3]
```

#### AREA\_Instrucoes

```

;          CODIGO SUBTRACAO SUCESSIVA PARA
DIVISAO
mov        R4,#0 ; i=0
ADD        R5,R1,#0 ; variavel aux que recebe o
valor de R1
LOOP1
CMP        R2,R1 ; verifica se R2 > R1
BGT        done1
SUB        R4,R1,R2 ; subtracao= R1-R2
ADD        R1,R4,#0 ; dividendo= R4 que no caso
eh o resto
add        R3,R3,#1 ; i++
B          LOOP1 ; volta para a condicao acima
done1
```

```

;CODIGO VERIFICA SE EH PAR OU IMPAR
CMP        R4,#0 ; verifica se o resto da
divisao eh 0
BNE        diferente ; se for diferente entao
salta para outra instrucao
MOV        R6,#0 ; K=0
MOV        R7,#0 ; variavel soma
```

#### MULTIPLICACAO

```

;          CODIGO SOMA SUCESSIVA PARA
LOOP2
CMP        R5,R6 ; verifica se R5 eh menor que
R6
BLE        DONE2 ; desvia o fluxo
ADD        R7,R7,R5 ; soma+=aux
ADD        R6,R6,#1 ; k++
```

	B	LOOP2 ; volta para a condicao acima
diferente	ADD	R8,R5,#1 ; caso for diferente
R8=aux+1		
	MOV	R6,#0 ; k=0
	MOV	R7,#0 ; usando a mesma variavel soma
	MOV	R10,#2 ; variavel constante 2
LOOP3		
	CMP	R10,R6 ; verifica se R10<R6
	BLE	DONE3 ; desvia o fluxo
	ADD	R7,R7,R8 ; soma+=R8
	ADD	R6,R6,#1 ; k++
	B	LOOP3 ; volta para a condicao acima
DONE2	ADD	R0,R7,#2 ; PAR -> adiciona em
R0=(numero*numero)+2		
	B	PAR
DONE3	ADD	R0,R7,#0 ; IMPAR -> adiciona em
R0=soma no caso (numero+1)*2		
	B	IMPAR
PAR	MOV	LR,R0
IMPAR	MOV	LR,R0

#### Questão 04

AREA_Variaveis		
num1	DCD	1000 ; variavel1 definida com esse
valor		
num2	DCD	70000 ; variavel2 definida com esse
valor		
	ADR	R1,num1
	LDR	R1,[R1]
	ADR	R2,num2
	LDR	R2,[R2]
AREA_Instrucoes		
	cmp	R1,R2 ; verifica se R1<=R2
	BLS	ELSE
	MOV	LR,R2 ; guarda em LR caso R2 caso eh
menor igual		
	B	FIM ; desvia o fluxo
ELSE		
	MOV	LR,R1 ; guarda em LR caso R1 eh menor
ou igual		
FIM ; fim da execucao		

### Questão 05

```
AREA_Variaveis
vetSalarios      DCD      100,100,100,100,100,100,100 ;
conteudo do vetSalarios
total            DCD      0
numFuncionarios DCD      7
                ADR      R1,vetSalarios
                ADR      R2,total
                LDR      R2,[R2]
                ADR      R3,numFuncionarios
                LDR      R3,[R3]

AREA_Instrucoes

                MOV      R4,#0 ; i=0
                MOV      R5,#0 ; k=0 que sera o iterador do
vetSalarios
LOOP
                LDR      R6,[R1,R5] ;acessa o conteudo do
vetor, pc+4
                CMP      R3,R4 ; verifica se 7 e maior que i
                BLE      FIM ; desvia se condicao for
satisfeita
                ADD      R2,R2,R6 ; total+=vetSalarios[i]
                ADD      R4,R4,#1 ; pc+4
                ADD      R5,R5,#4 ; itera de 4 em 4 bytes
                B         LOOP

FIM
                MOV      LR,R2 ; guarda o valor de R2 no
registrador LR
```

### Questão 07

```
vet      fill      400 ; vetor de TAM 100, considerando que o cada
posicao o tamanho eh 4 bytes
array    DCD      0 ; representando o array do exemplo com
tamanho fixo
                ADR      R0,vet

preenche
                MOV      R2,#0 ;i=0
                MOV      R3,#0 ; PC+4
                MOV      R4,#0 ; VET[X]
                MOV      R5,#0 ; K=0
                MOV      R6,#0 ; S
                MOV      R7,#0 ; PC2+4

LOOP
                CMP      R2,#100 ; Verifica se i>100
```

```

        BGE      LOOP2 ; desvia o fluxo, caso a condicao nao
for satisfeita
        STR      R4,[R0,R3] ; Adiciona na posicao Vet[x]=R4
        ADD      R2,R2,#1 ; i++
        ADD      R4,R2,#0 ; R4=i
        ADD      R3,R3,#4 ; PC+4
        B        LOOP ; Volta para a condicao acima

LOOP2
        CMP      R5,#100 ; verifica se k>100
        BGE      FIM ; desvia o fluxo, caso a condicao nao for
satisfeita
        LDR      R4,[R0,R7] ; Pega o elemento da posicao vet[x]
        ADD      R6,R6,R4 ; S=S+VET[X]
        ADD      R5,R5,#1 ; k++
        ADD      R7,R7,#4 ; PC+4
        B        LOOP2 ; Volta para a condicao acima

FIM
        MOV      LR,R6 ; Coloca em LR a soma de S=S+VET[X]

```

### Questão 08

```

AREA_Variaveis
vet1      DCD      1,2,3 ; vetA
vet2      DCD      3,2,1 ; VetB
          ADR      R1,vet1
          ADR      R2,vet2

          mov      R3,#0 ; soma=0
          mov      R4,#0 ; i=0
          MOV      R6,#0 ; PC+4

AREA_Instrucoes

LOOP1
          CMP      R4,#3 ; i< TAM
          BGE      done1
          LDR      R7,[R1,R6] ; VetA[j]
          mov      R5,#0 ; j=0;

LOOP2
          CMP      R5,R7 ; j<vetA[j]
          BGE      done2
          LDR      R8,[R2,R6] ; vetB[j]
          ADD      R3,R3,R8 ; soma+=vetB[j]
          ADD      R5,R5,#1 ; j++

```

LOOP2	B	LOOP2 ; Volta para a condicao acima
done2	ADD	R4,R4,#1 ; i++
	ADD	R6,R6,#4 ; PC+4
	B	LOOP1 ; volta para a condicao acima
LOOP1		
done1	MOV	LR,R3 ; Mostra o resultado em LR da
soma		

### Questão 09

AREA_Variaveis		
ant	DCD	1 ; variavel com o valor anterior
atual	DCD	1 ; variavel com o valor atual
soma	DCD	0 ; variavel que vai somar a serie
	ADR	R0,ant
	LDR	R0,[R0]
	ADR	R1,atual
	LDR	R1,[R1]
	ADR	R2,soma
	LDR	R2,[R2]
AREA_Instrucoes		
	MOV	R3,#3 ; i=3, pois a iteracao ate 3 o
valor nao muda que eh 1		
LOOP		
	CMP	R3,#10 ; verifica se 3 eh menor ou
igual a 3		
	BGT	DONE
	ADD	R2,R0,R1 ; soma=ant+atual
	ADD	R0,R1,#0 ; ant=atual
	ADD	R1,R2,#0 ; atual=soma
	ADD	R3,R3,#1 ; i++
	B	LOOP ; volta para condicao acima
DONE		
	MOV	LR,R2 ; coloca em LR o resultado da
serie de n=10 LR=55		

### Questão 10

```
AREA_Variaveis
xa          DCD      5 ; valor da variavel que sera o
multiplicando
ya          DCD      4 ; valor da variavel que sera o
multiplicador

          ADR      R1,xa
          LDR      R1,[R1]
          ADR      R2,ya
          LDR      R2,[R2]

AREA_Instrucoes

          mov      R3,#0 ; i=0
          mov      R4,#0 ;soma=0

LOOP

          CMP      R2,R3 ; verifica se o ya<R3
          BLE      done ; desvia o fluxo
          add      R4,R4,R1 ; soma+=x no caso, como se
fosse soma=xa+xa+xa...
          add      R3,R3,#1 ; i++
          B        LOOP ; volta para a condicao acima

done

          MOV      LR,R4 ; adiciona em LR o valor da
multiplicacao nesse ex. 20
```

### Questão 11

```
AREA_Variaveis
dividendo   DCD      9 ; dividendo
divisor     DCD      3 ; divisor como na formula
dividendo/divisor
quociente   DCD      0 ; variavel que guarda o resultado
da subtracao

          ADR      R1,dividendo
          LDR      R1,[R1]
          ADR      R2,divisor
          LDR      R2,[R2]
          ADR      R3,quociente
          LDR      R3,[R3]

AREA_Instrucoes

          mov      R4,#0 ;sub=0

LOOP

          CMP      R2,R1 ; verica se R2>R1
          BGT      done ; desvia o fluxo
          SUB      R4,R1,R2 ; sub=dividendo-divisor
          ADD      R5,R1,#0 ; resto=dividendo
```

```

                                ADD      R1,R4,#0 ; dividendo=sub
                                add      R3,R3,#1 ; quociente++
                                B        LOOP
done
                                MOV      LR,R3 ; coloca em LR o valor da
subtracao que no caso eh o quociente

```

### Questão 12

```

vet      DCD      7,5,2,1,1,3,4 ; Vetor
        ADR      R0,vet

        MOV      R1,#1 ; k=1

LOOP1
        CMP      r1,#7 ; k<n
        BGE      done1 ;

        MOV      R2,#0 ; j=0
        MOV      R3,#0 ; aux=0
        MOV      R4,#0 ; PC+4
        MOV      R5,#4 ;PC2+4

LOOP2
        CMP      r2,#6 ;j<n-1
        BGE      done2
        LDR      R6,[R0,R4] ; VET[j]
        LDR      R7,[R0,R5] ; VET[J+1]

        CMP      R6,R7 ; vetor[j]>vetor[j+1]
        BLE      done3
        ADD      R3,R6,#0 ; aux=vetor[j]
        STR      R7,[R0,R4] ; vetor[j]=vetor[j+1]
        STR      R3,[R0,R5] ; vetor[j+1]=aux

        ADD      R2,R2,#1 ; j++
        ADD      R4,R4,#4 ; PC1+4
        ADD      R5,R5,#4 ; PC2+4

done3    B        LOOP2 ; volta para a condicao acima LOOP2
        ADD      R2,R2,#1 ; senao j++

```



```
done2      ADD      R1,R1,#1 ; k++
           B        LOOP1 ; volta para a condicao acima LOOP1

done1
vetor[i]   LDR      LR,[R0,#0] ; Mostra em LR o conteudo do
```