

## Primeira Lista de Exercícios

1ª Questão) Traduza o seguinte loop para C. Assuma que o inteiro i é armazenado no registrador \$t1, \$s2 armazena o inteiro chamado result, e \$s0 armazena o endereço base do inteiro MemArray

```
addi $t1, $0, $0
LOOP: lw $s1, 0($s0)
add $s2, $s2, $s1
addi $s0, $s0, 4
addi $t1, $t1, 1
slti $t2, $t1, 100
bne $t2, $s0, LOOP
```

2ª Questão) Considere o seguinte loop em MIPS

```
LOOP: slt $t2, $0, $t1
beq $t2, $0, DONE
subi $t1, $t1, 1
addi $s2, $s2, 2
j LOOP
DONE:
```

a) Assuma que o registrador \$t1 é inicializado com o valor 0. Qual é o valor no registrador \$s2 assumindo que \$s2 inicialmente possui valor zero.

b) Para o código escrito em assembly MIPS, escreva a rotina equivalente em C. Assuma que os registradores \$s1, \$s2, \$t1 e \$t2 são inteiros A, B, i e temp, respectivamente.

3ª Questão) Traduza o seguinte código em C para assembly MIPS. Use o número mínimo de instruções. Assuma que os valores de a, b, i e j estão armazenados nos registradores \$s0, \$s1, \$t0 e \$t1 respectivamente. Também, assumo que registrador \$s2 armazene o endereço base da vetor D

```
for (i=0; i<a; i++)
for (j=0; j<b; j++)
D[4*j] = i + j;
```

4ª Questão) Descreva detalhadamente o que cada código faz

```
.text
main:
li $v0, 5
syscall
move $t0, $v0
li $v0, 5
syscall
move $t1, $v0
bgt $t0, $t1, t0_bigger
move $t2, $t1
b endif
t0_bigger:
move $t2, $t0
endif:
move $a0, $t2
li $v0, 1
syscall
li $v0, 10
syscall
```

```
.text
.globl main
main:
li $a0, 5
jal Funcao
move $s0, $v0
li $v0, 10
syscall
Funcao:
sub $sp, $sp, 4
sw $ra, 0($sp)
li $t1, 1
slti $t0, $a0, 2
beq $t0, $zero, Calcula
add $v0, $zero, $zero
beq $a0, $zero, Sai
add $v0, $t1, $zero
Sai:
lw $ra, 0($sp)
add $sp, $sp, 4
jr $ra
Calcula:
```

	<pre> add \$a1, \$a0, \$zero Loop: sub \$a1, \$a1, \$t1 jal Multiplica add \$a0, \$v0, \$zero bne \$a1, \$t1, Loop j Sai Multiplica: mult \$a0, \$a1 mflo \$v0 jr \$ra </pre>
--	---

5ª Questão) Escreva um programa em linguagem de montagem (assembly language) do processador MIPS que processa um vetor de números inteiros (VET), transformando cada elemento do vetor em seu valor absoluto. Por exemplo, o valor absoluto do número -43 é +43 e do número +55 é +55. O programa deve obrigatoriamente chamar uma sub-rotina calc\_abs que recebe como parâmetro um número e calcula o valor absoluto deste, retornando-o segundo as convenções do MIPS. O valor retornado pela sub-rotina deve ser armazenado, pelo programa principal, na mesma posição do array que continha o elemento original. Utilize a área de dados abaixo para escrever o seu programa.

**.data**

**VET: .word 23 -43 55 -9 -7 21 -76 12 -45 -10**

**TAM: .word 10**

6ª Questão) Escreva um programa em linguagem de montagem (assembly language) do processador MIPS que processa um vetor de números inteiros (VET), contando quantos elementos pares e ímpares contém o vetor. Os resultados devem ser armazenados nas variáveis PAR e IMPAR em memória. Utilize a área de dados abaixo para escrever o seu programa. Implemente sua funcionalidade com instruções reais da arquitetura.

**.data**

**VET: .word 23 43 55 9 7 21 76 12 45 10**

**TAM: .word 10**

**PAR: .word**

**IMPAR: .word**

**.text**

7ª Questão) (2,5 pontos) Apresente a implementação em linguagem de máquina dos códigos abaixo

**Enquanto i<100**

**v[i+1] \_ v[i] +1;**

**se v[i+1] <> 10**

**a++;**

**fim se;**

**b ← c\*2 -d;**

**fim enquanto**

**int fact(int n)**

**{**

**if (n < 1)**

**return 1;**

**else**

**return ((n+1)\*fact(n-1));**

**}**

**\* função fact passagem de parâmetros pelo registrado \$a0**

\* a, b, c, d, i estão nos registros  $\$t_0$ ,  $\$t_1$ ,  $\$t_2$ ,  $\$t_3$  e  $\$t_4$

8ª Questão) As arquiteturas de uso geral atuais são normalmente referenciadas como máquinas von Neuman. Quais as principais características dessa arquitetura e sua principal limitação?