Universidade Tecnológica Federal do Paraná
Campus Campo Mourão
Curso: Bacharelado em Ciência da Computação
Disciplina: Arquitetura e Organização de Computadores
Prof. Dr. João Fabrício Filho

Os exercícios foram adaptados do livro
Patterson, David A. Hennessy, John L. Organização e Projeto de Computadores. Disponível em: Minha Biblioteca, (5a. edição). Grupo GEN, 2017.

1 <§2.7> Traduza o código C para o código assembly do MIPS. Use um número mínimo de instruções. Suponha que os valores de a, b, i e j estejam nos registradores $s0, $s1, $t0, $t1, respectivamente. Além disso, suponha que o registrador $s2 mantenha o endereço de base do array D.

a)

```
if (a < b) {
   i = j;
} else {
   i = 0;
}
```

b)

```
if (D[i] < b) {
   D[i] += a;
}
```

c)

```
while (D[i] != 0) {
   i++;
}
```

d)

```
if (D[i] < a) {
   D[j] += b;
} else {
   D[i] += b;
}
```

e)

```
if (D[i+j] < D[i]) {
   D[i]   += D[i+j];
   D[i+j] =  D[i] - D[i+j];
} else {
   D[i]   -= D[i+j];
}
```

f)

```
while (D[i+j]<a)
   D[i+j] += b;
```

g)

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

h)

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

i)

```
while (D[i] != 0) {
   i++;
}
```

j)

```
for (i = 0; i < a; i++) {
   for (j = 0; j < b; j++) {
      D[i + j] = i - j;
   }
}
```

2 <§2.7> Traduza os laços de repetição abaixo para linguagem de alto nível. Suponha que o inteiro i seja mantido no registrador $t1, $s2 mantenha a variável chamada result, e $s0 mantenha o endereço de base do vetor MemArray.

a)

```
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, $0, LOOP
```

b)

```
LOOP:
   sll  $t0, $t1, 2
   add  $t0, $t0, $s0
   lw   $t2, 0($t0)
   beq  $t2, $s2, EXIT
   sll  $t2, $t2, 1
   sw   $t2, 0($t0)
   addi $t1, $t1, 1
   j LOOP
EXIT:
```

c)

```
addi $t1, $0, $0
addi $s2, $0, $0
LOOP:
   lw $s1, 0($s0)
   slt $t2, $s3, $s1
   add $s2, $s2, $t2
   addi $s0, $s0, 4
   addi $t1, $t1, 1
   slti $t2, $t1, 100
   bne  $t2, $0, LOOP
```

d)

```
addi $t1, $0, $0
LOOP:
   lw $s1, 0($s0)
   beq $s1, $0, EXIT
   slti $t2, $s1, 0
   beq $t2, $0, POSITIVO
   sw $s1, 0($s3)
   j PROXIMO
POSITIVO:
   sll $t2, $s1, 1
   sw $t2, 0($s3)
PROXIMO:
  addi $s0, $s0, 4
  addi $s3, $s3, 4
  addi $t1, $t1, 1
  slti $t2, $t1, 100
  bne $t2, $0, LOOP
```