CSC 225 - Computer Architecture/Assembly Language

Assignment #3

Name: Christopher Welch **Score: ___/25**

Part I - Directions: 19 Points (1 Point each: Using Appendix A, translate each of the following Pseudocode expressions into MIPS assembly language – (a) through (s) on pages 22 and 23.

Part II – Directions: 6 Points - Page 23 – Per the instructions, please show your work!

---

a) t3 = t4 + t5 – t6;

```
add $t4, $t4, $t5  # t4 + t5
sub $t4, $t4, $t6  # (t4+t5) – t6
li $t3, $t4        # t3 = t4
```

b) s3 = t2 / (s1 – 54321);

```
sub $s1, $s1, 54321    # s1 – 54321
div $t2, $s1           # t2 / (s1 -54321)
mflo $s3               # s3 = t2 / s1
```

c) sp = sp – 16;

```
sub sp, sp, 16     # sp = sp – 16;
```

d) cout << t3;

```
li $t3, 1       # System call code for print_int
syscall         # Prints integer onto stdout (the screen).
```

e) cin >> t0;

```
li $t0, 5       # system call code for read_int
syscall         # reads input value and places it in $t0
```

f) a0 = &array;

```
la $a0, &array      # Loads $a0 with a pointer to the address of &array
```

g) t8 = Mem(a0);

```
la $t8, $a0  # Loads $t8 with a pointer to the address of the $a0 array
```

h) Mem(a0+ 16) = 32768;

```
li $t0, 32768        # $t0 = 32768      $t0 = temporary variable.
lw $t0, 16($a0)    # $t0 = MEM[$a0+16]
```

i) cout << “Hello World”;

```
Hw:   .asciiz “Hello World”   # Hw is a label for the string “Hello World”
      li $v0, 4               # System call code for Print String
      la $a0, Hw              # loads address of Hw into $a0
      syscall                 # prints the string attached to Hw
```

j) If (t0 < 0) then t7 = 0 – t0 else t7 = t0;

```
      bgez $t0, else          # If (t0 < 0), branch to else
      sub $t7, $zero, $t0     # t7= 0 – $t0
      b next                  # branch around the else code to the next code
else: la $t7, $t0             # t7 = t0
      next:
```

k) while (t0 != 0) { s1 = s1 + t0; t2 = t2 + 4; t0 = Mem(t2) };

```
      while:
      beq $t0, $zero, done    # if ($t0 = 0), branch to done
      add $s1, $s1, $t0       # $s1 = $s1 + $t0
      addi $t2, $t2, 4        # $t2 = $t2 + 4
      la $t0, $t2             # Loads $t0 with a pointer to the address of $t2
      b while                 # branch to while
      done:                   # next part of code, when $t0 = 0
```

l) for (t1 = 99; t1 > 0; t1=t1 -1) v0 = v0 + t1;

```
li $t1, 99                  # $t1 = 99
loop:
add $v0, $v0, $t1           # v0 = v0 + t1
addi $t1, $t1, -1           # t1=t1 -1
bgtz $t1, loop              # if ($t1 > 0), branch to loop
```

m) t0 = 2147483647 – 2147483648;

```
li $t1, 2147483647          # $t1 = 2147483647
li $t2, 2147483648          # $t1 = 2147483648
sub $t0, $t1, $t2           # $t0 = $t1 - $t2
```

n) s0 = -1 * s0;

```
mult -1, $s0          # -1 * s0
mflo $s0              # $s0 = Lower 32-bits of the product
```

o) s1 = s1 * a0;

```
mult $s1, $a0         # s1 * a0
mflo $s1              # s1 = (s1 * a0)
```

p) s2 = srt(s0*s0 + 56) / a3;

```
mult $s0, $s0       # $s0 * $s0
mflo $t0            # $t0 = $s0 * $s0
addi $t0, 56        # $t0 + 56
multi $t0, $t0      # $t0 * $t0
mflo $t0            # $t0 = $t0 * $t0
div $t0, $a3        # $t0 / $a3
mflo $s2            # $s2 = $t0 / $a3
```

q) s3 = s1 – s2 / s3;

```
div $s2, $s3              # $s2 / $s3
mflo $t0                  # $t0 =  $s2 / $s3
sub $s3, $s1, $t0         # $s3 = $s1 - $t0
```

r) s4 = s4 * 8;

```
mult $s4, 8         # s4 * 8
mflo $s4            # s4 = s4 * 8
```

s) s5 = 7 * s5;

```
mult $s5, 7         # 7 * s5
mflo $s5            # s5 = 7 * s5
```

The assignment should be completed using and submitted in either a .rtf, .doc, or .pdf. Type the question, and then the answer.

When you have finished the assignment, please upload the answers to the Assignment #3 Dropbox on D2L.

Spelling and grammar will count! ½ Point Per Error