COMP 122/L Practice Exam P2 (Answers)

This is representative of the kinds of topics and kind of questions you mav be asked on the midterm. In addition to this practice exam, you should also review labs

1.) What will the following program print, if run with SPIM?

```
li $a0, 83
li $v0, 1
syscall
```

83

2.) What does the following program print, if run with SPIM?

```
li $a0, 15
li $v0, 1
syscall
li $a0, 'a'
li $v0, 11
syscall
li $a0, 4
li $v0, 1
syscall
```

15a4

3.) What does the following program print, if run with SPIM?

```
li $a0, 24
li $v0, 1
syscall
li $a0, 47
li $v0, 1
syscall
```

2447

4.) What does the following program print, if run with SPIM?

```
.data
foo:
    .asciiz "Some string\n"
bar:
    .asciiz "Some other string\n"
main:
  la $a0, bar
  li $v0, 4
  syscall
  li $v0, 10
  syscall
```

```
Some other string
```

5.) What does the following program print, if run with SPIM?

```
.data
foo:
    .ascii "alpha"
bar:
    .asciiz "beta"
main:
  la $a0, bar
  li $v0, 4
  syscall
  li $v0, 10
  syscall
```

```
beta
```

6.) What does the following program print, if run with SPIM, and 4 is input by the user?

```
li $v0, 5
syscall
addiu $a0, $v0, 3
li $v0, 1
syscall
```

```
7
```

7.) Convert the following C-like code into MIPS assembly. The names of the variables reflect which registers must be used for the MIPS assembly. Do not assume any initial values for the registers. You may use additional registers.

```
$t0 = 3;
$t1 = 7;
$t2 = ($t0 * $t1) + 8;
```

```
li $t0, 3
li $t1, 7
mult $t0, $t1
mflo $t2
addiu $t2, $t2, 8
```

8.) Convert the following C-like code into MIPS assembly. The names of the variables reflect which registers must be used for the MIPS assembly. Do not assume any initial values for the registers. You may use additional registers.

```
int s0 = 82;
int s1 = s0 << 2;
int s2 = s1 * 20;
int s3 = s2 + 7;
int s4 = s3 - 24;
int s5 = s4 / 3;
```

```
main:
  li $s0, 82          # int s0 = 82;

  sll $s1, $s0, 2     # int s1 = s0 << 2;

  li $t0, 20          # int s2 = s1 * 20 (part 1 of 3)
  mult $s1, $t0       #                  (part 2 of 3)
  mflo $s2            #                  (part 3 of 3)

  addi $s3, $s2, 7    # int s3 = s2 + 7

  li $t1, 24          # int s4 = s3 - 24 (part 1 of 2)
  sub $s4, $s3, $t1   #                  (part 2 of 2)

  li $t2, 3           # int s5 = s4 / 3 (part 1 of 3)
  div $s4, $t2        #                 (part 2 of 3)
  mflo $s5            #                 (part 3 of 3)
```

9.) Convert the following C-like code into MIPS assembly. The names of the variables reflect which registers must be used for the MIPS assembly. Do not assume any initial values for the registers. You may use additional registers. The portions in <<>> will require you to use QtSpim functionality. You do not need to exit the program properly.

```
int s0 = <<read integer from the user>>;
int s1 = s0 + 3;
<<print integer s1>>
```

```
main:
  li $v0, 5
  syscall
  move $s0, $v0
  addi $s1, $s0, 3

print:
  li $v0, 1
  move $a0, $s1
  syscall
```

10.) Convert the following C-like code into MIPS assembly. The names of the variables reflect which registers must be used for the MIPS assembly. Do not assume any initial values for the registers. You may use additional registers.

```
$t0 = 3;
$t1 = 7;
$t2 = ($t0 * $t1) + 8;
```

```
li $t0, 3
li $t1, 7
mult $t0, $t1
mflo $t2
addiu $t2, $t2, 8
```