EGC442 Problem Set 7 Dr. Izadi

First Name:_________________________ Last Name: ___________________________

1) A main program will call a procedure Power for computing x^y. Currently, x is in $s0, y is in $s1. How might the program pass the parameter values to Power?

○ add $a0, $s0, $zero
add $a1, $s1, $zero

○ add $s0, $a0, $zero
add $s1, $a1, $zero

○ add $v0, $s0, $zero
add $v1, $s1, $zero

2) A first part of a main program calls procedure Power to compute x^y, where x is in $s0, y is in $s1. Later, the program is to call Power again, but this time x is in $s3 and y is in $s7. How might the program pass the parameter values to Power?

○ Copy $s3 to $a0, and $s7 to $a1.

○ Not possible; x and y must be in $s0 and $s1.

3) A main program calls a Power procedure using the instruction: jal Power. That instruction is at address 1000. What happens to $ra?

○ Nothing; jal is unrelated to $ra.

○ $ra is set to 1000.

○ $ra is set to 1004.

4) A procedure Power computes $a0 to the power of $a1. In which register should Power write the result before returning?

○ $a0

○ $v0

○ $s0

5) A procedure Power computes $a0 to the power of $a1. How should the procedure jump back to the next instruction in the caller?

○ jr Caller

○ jr $ra

○ jal $ra

6) The stack is a region in the set of registers.

○ True

○ False

7) The jal instruction copies registers to the stack.

○ True

○ False

8) A procedure should copy all of registers $t0-$t9 and $s0-$s7 to the stack, before executing the procedure's computations.

○ True

○ False

9) If a procedure will update registers $s0, $s1, $s2, and $s3, the procedure should make room on the stack by adding 16 to $sp.

○ True

○ False

10) Upon computing a value to return, the procedure might copy that value into register $v0.

○ True

○ False

11) MIPS allows a procedure to modify registers $t0-$t9 without saving those registers to the stack and restoring those registers upon returning.

○ True

○ False

12) Write a MIPS subroutine to carryout the following function.

```
temp = v[k];
v[k] = v[k + 1];
v[k + 1] = temp;
```

Assume base address of v is register $a1, k is in register $a2, and temp is assigned to $s1.