Name: (as it would appear on official course roster)
UCSB email address: **@ucsb.edu**
Lab Section Time:
Optional: name you wish to be called if different from above
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")

Assignment 04: MIPS Instructions

Assigned: *Friday, January 31st, 2020*
Due: *Wednesday, February 5th, 2020*
Points: *80 (normalized to 100 in gradebook)*

- You may collaborate on this homework with AT MOST one person, an optional "homework buddy".
- MAY ONLY BE TURNED IN ON **GRADESCOPE as a PDF file** (see instructions in online lab01 description).
- There is NO MAKEUP for missed assignments.
- We are strict about enforcing the LATE POLICY for all assignments (see syllabus).

Only use the space provided for answers. Use clear and clean handwriting (or typing). You will get penalized if you are asked to show your calculations and do not do so.
ALWAYS SHOW YOUR WORK!

1. (6 pts) Assume the following register contents:

   `$t0 = 0xAAAAAAAA, $t1 = 0x12345678`

   a. (2 pts) What is the value of `$t2` for the following sequence of instructions?

   ```
   sll $t2, $t0, 44
   or $t2, $t2, $t1
   ```

   b. (2 pts) Same question, but for these instructions?

   ```
   sll $t2, $t0, 4
   andi $t2, $t2, -1
   ```

   c. (2 pts) Same question, but for these instructions?

   ```
   srl $t2, $t0, 3
   andi $t2, $t2, 0xFFEF
   ```

Name:

(as it would appear on official course roster)

2. (4 pts) Assume the following register contents:

   `$t0 = 0x00101000, $t1 = 0x12345678`

   What is the value of `$t2` for the following sequence of instructions?

```
      slt $t2, $zero, $t0
      bne $t2, $zero, else
      j done
else: addi $t2, $t2, 2
done: ori $v0, $zero, 10
      syscall
```

3. (6 pts) What is the value of the following hex numbers in decimal if (a) they represent two's complement numbers, and (b) they represent an unsigned integer?

   a. (3 pts) n = **0x0C000000**

   b. (3 pts) n = **0xD000000D**

Name:

(as it would appear on official course roster)

4. (12 pts) Translate the following MIPS machine code instruction into MIPS assembly. Use the MIPS Reference Card to help you decide your answers.
   a. (4 pts) **0x0C000000**

   b. (4 pts) **0x11110002**

   c. (4 pts) **0x00402020**

Name:

(as it would appear on official course roster)

5. (12 pts) Consider the following MIPS assembly code shown below with line numbers. You are told that the first instruction (line 0) resides in MIPS memory address **0x00440000**.

```
0:          lw $t1, 4($t0)
1:          ori $t2, $zero, 5
2:          ori $t3, $zero, $zero
3:    loop: add $t1, $t2, $t1
4:          beq $t1, $t3, exit
5:          addi $t1, $t1, 1
6:          addi $t2, $t2, -3
7:          and $t2, $t1, $t2
8:          j loop
9:    exit: addi $v0, $zero, 10
10:         syscall
```

a. (4 pts) What is the MIPS machine instruction in hexadecimal on line 0?

b. (4 pts) What is the MIPS machine instruction in hexadecimal on line 4?

c. (4 pts) What is the MIPS machine instruction in hexadecimal on line 8?

Name:

(as it would appear on official course roster)

6. (20 pts) Write a full MIPS assembly language program that implements the following C++ code *without using* any pseudocodes AND using hexadecimal numbers when using immediate values. Using **`syscall`** is allowed. Save your program as **Lab04Q6.asm** and submit on Gradescope.

```
int rick(int m, int n) {
      return (2*m - n);
}
int main() {
      int b = 2, v;
      int a[5] = {3, 5, 2, 1, 7};
      for (int i = 0; i < 5; i++) {
            if (a[i] > b) v = rick(a[i], b);
            else v = 0;
            cout << v << " units.\n";
      }
      return(0);
}
```

7. (20 pts) Write a full MIPS assembly language program that implements the following C++ code *without using* any pseudocodes AND using hexadecimal numbers when using immediate values. *In addition, make sure you adhere to the MIPS Calling Convention*. Using **`syscall`** is allowed. Save your program as **Lab04Q7.asm** and submit on Gradescope.

```
int jerry(int x) {
      return 3*(x+2);
}
int morty(int y) {
      int m = jerry(2*y - 6);
      int n = jerry(m);
      return n + m;
}
int main() {
      int v = morty(5);
      cout << v*4;
      return(0);
}
```