CSCI 2500 — Computer Organization
Lab 06 (document version 1.1)

- This lab is due by the end of your lab session on Wednesday, October 10, 2018.
- This lab is to be completed **individually**. Do not share your code with anyone else.
- You **must** show your code and your solutions to a TA or mentor to receive credit for each checkpoint.
- Labs are available by 6:00PM on Mondays before your lab sessions. Plan to start each lab early and ask questions during office hours, in the discussion forum on Submitty, and during your lab session.

1. **Checkpoint 1:** For the first checkpoint, download the `bit.s` MIPS code example from the `lec-10-05` folder on Submitty. Understand how the code works to produce the output below:

```
-------------x----------------
-x----------xxx---------------
-xx----------x--x--------------
x--x---------x---x-------------
x---x--------x---x-------------
x----x-------x----xxxxxx--------
x-----x------x----------xx------
x------xx---x--xx--------xxxxxx-
x-----x---xx--xxxx-----------xx
-x---x---xx----xx---------xxxxx-
--x--x--x---x-----x---xxxx-x----
---x-x-x-------------xxxx-x----
xxx--xx---------xxxx------x-----
x---x--x---------xx----x--x-----
-x---x-x-----------------x------
--x--x-x---------------xx------
---x-xx-------------------xx----
----xx-----------------xx---x---
----x----------------x--x---x--
---x-----------------x----xxxx--
--x-------------------x---------
--x--------------------x--------
--x-----xxxxxxxx-------x--------
---x--xx--------xx----x---------
---x--x-----------x----xx-------
--xxxx------------xxxxxxxx-----
```

   Modify the given code to eliminate the "`j merge`" instruction in the `printbit` procedure. In other words, revise the procedure to use only one branch instruction.

   To receive credit for this checkpoint, you need to show both your code revision and successful output.

2. **Checkpoint 2:** Similar to Checkpoint 1, revise the `bitcount` procedure in `bit.s` to eliminate the `beq` instruction. More specifically, you can only use the one `bne` instruction and the one `jal` instruction; no other branch or jump instructions are allowed.

   Try running your code using the hexadecimal words shown in the `lab06a.txt` and `lab06b.txt` files available in Submitty.

   And to test, add a `syscall` to `print_int` to display the return value (i.e., temporary register `$t0`) before you print the newline and return.

   To receive credit for this checkpoint, you need to show both your code revision and successful output.

3. **Checkpoint 3:** For the third checkpoint, make a copy of the revised `bit.s` code, then modify this code as described below.

   Currently, this code displays each 32-bit word of the given data on a line by itself, i.e., one word per line. Extend the `bitcount` procedure to display n words per line, where n is specified as the third argument to `bitcount` in register `$a2`.

   Test your code by setting `$a2` to 1. Then test your code using the `lab06b.txt` file with `$a2` set to 2. What does your output look like?

   As above, to receive credit for this checkpoint, you need to show both your code revision and successful output.