

## Cosc 2P12

### Assignment 3

Due date: Friday Nov. 11<sup>th</sup> 17:00

Late date: Monday November 14<sup>th</sup>, 17:00

#### Part A

Write a MIPS program that prompts for and reads in a positive signed 32-bit integer **n** (use the `print_sring` and `read-int` syscalls). It should then calculate and print out the binary representation of that integer, one digit at a time, from the most significant to the least bit. Since we know it will not be greater than a 31 bit number (since it is positive integer) you should print a 0 first to make it a full 32 bit binary representation.

For example, if the user entered 3072015 it should look like the following:

```
Please enter a positive 32-bit integer: 3072015
0000000001011101110000000001111
```

#### Part B

Modify your program from **part A** to add a loop when asking for the integer **n** and continue to loop until the user enters a 0. If they provide a negative number, you should print an error message and loop again. You will want to print a newline character after each loop to ensure that the output is easy to read.

#### Part C

Modify your program from **part B** such that, instead of just printing the digits immediately to the screen, your program will store the digits (including the leading 0) into a character array and then print that string when you are done processing the number. Since we know it is a 31 bit number (since it is positive), we can define our output array as either `.space 33` or `.asciiz` with a statically assigned 32 character string of zeros (remember: the `print_string` syscall requires a null terminated character array/string). So the output will be identical as **Part A** for each number retrieved from the user.

Test your program with several values of **n**. Provide output for `n=1023`, `20480` and `30000000`

For each MIPS program be sure to use proper documentation to describe what the code is doing. See labs and lecture videos for proper comments and formatting.

## **Submission**

This assignment must be submitted electronically using Sakai by the date above (note all times are in EST). Include the MIPS assembly files (one for each part) and a text file of the test output.

The TA will be running your program to ensure it is fully functional. Make the marker happy!!!

For the electronic submission, use Sakai, an assignment 3 submission will be available.

For the electronic submission using Sakai please double check that you included:

- Three MIPS .asm files (one for each part).
- Text file demonstrating your test output.
- Package above into a zip file and submit.

The End