Assignment 4

Due: 9:00am, Thu Feb 15th, 2018

Note: Make reasonable assumptions where necessary and clearly state them. Feel free to discuss problems with classmates, but the only written material that you may consult while writing your solutions are the textbook and lecture slides/videos/notes.

For the MARS programs, please upload separate .asm files that can be easily tested by the TAs. Note that your MARS programs will be graded on readability and user friendliness, as well as correctness. That means LOTS OF COMMENTS!! Again, here's the document that provides an overview of MARS. The three questions are worth 20, 30, and 50 points.

1. Read the assembly code below and explain what this procedure is trying to do in 1-2 sentences. Also add a comment for each assembly instruction. Hints: $a0 points to an array of sorted integers (smallest first). When new-proc is called for the first time, assume that $a3 is 0, $a1 is 57, and $a2 is 16.

```
new_proc:
  beq $a2, $zero, proc_fail
  srl $a2, $a2, 1
  add $t0, $a2, $a3
  sll $t1, $t0, 2
  add $t2, $a0, $t1
  lw $t3, 0($t2)
  beq $t3, $a1, proc_succ
  bgt $t3, $a1, call_again
  add $a3, $zero, $t0
call_again:
  addi $sp, $sp, -4
  sw $ra, 0($sp)
  jal new_proc
  lw $ra, 0($sp)
  addi $sp, $sp, 4
  jr $ra
proc_succ:
  add $v0, $zero, $t0
  jr $ra
proc_fail:
  addi $v0, $zero, -1
  jr $ra
```

2. **Next Square:** Write a MIPS assembly program in the MARS simulator that solves the following problem. Given a number N and its square (N^2), the square of N+1 can be computed with the following equation: (N+1)^2 = N^2 + 2*N + 1 = N^2 + N + (N + 1). In other words, you can compute the square of N+1 by adding N and N+1 to the square of N. Your program should have a main routine that (i) prompts the user to enter the values for N and N^2, (ii) reads in these two integer values and confirms they are greater than zero (print an error message and exit if this is not true), (iii)

enters a loop that prints the squares of numbers (N+i) from i=1 to i=3. The loop should call a procedure "nextsq" to implement the math. Procedure nextsq takes in arguments X and X^2 and returns the value X^2 + X + (X+1). You may use other procedures if you wish.

Here are some examples of how your program should behave:

- If the inputs to your program are 2 4, the output should be:
  ```
  Enter N and N^2 (both positive):
  2
  4
  Next 3 squares are: 9 16 25
  ```
- If the inputs to your program are -3 9, the output should be:
  ```
  Enter N and N^2 (both positive):
  -3
  9
  The input is erroneous.
  ```

3. **Compressed Genomic Data:** Write a MIPS assembly program in the MARS simulator that accepts an input string of size less than 40 characters, applies the following decompression algorithm to the string, and then prints the resulting decompressed string. In the input string, if a "#" is encountered, the next byte is interpreted as a number *i* between 0-255; the output string would then replace the # and its *i* with *i-32* consecutive occurrences of the character "A" (see examples below). If *i=32*, then the output string would replace the # and its *i* with one occurrence of "#". Similarly, "$" corresponds to multiple occurrences of the character "C"; "%" corresponds to multiple occurrences of the character "G"; "&" corresponds to multiple occurrences of the character "T". For all other encountered characters, the output string should simply reproduce that character. We will only test your code with valid inputs, i.e., strings of under 40 characters and *i > 31.* Keep an ASCII table handy. See the following examples:
   - Provide an input of less than 40 characters:
     ```
     4A2#+96$(XY%"TV&&p$ d
     The decompressed string is:
     4A2AAAAAAAAAAA96CCCCCCCCXYGGTVTTTTTTp$d
     ```
     Explanation: "#" is followed by "+", which is ASCII 43, so it was replaced by 11 A's. "$" is followed by "(", which is ASCII 40, so it was replaced by 8 C's. "%" is followed by double-quote, which is ASCII 34, so it was replaced by 2 G's. "&" is followed by "&", which is ASCII 38, so it was replaced by 6 T's. "$" is followed by " " (space), which is ASCII 32, so it was replaced by "$".
   - Provide an input of less than 40 characters:
     ```
     ab%%& G#)mn
     The decompressed string is:
     abGGGGG&GAAAAAAAAAmn
     ```