COE 147 Spring 2014
Lab 5 Solution: Using Stacks and Recursive Functions

Part 1: Using Stacks

```
.data

prompt:     .asciiz "Enter your string?\n"
message:    .asciiz "Here is the output?\n"
input_str:  .space 64

.text

#display prompt
la $a0,prompt
li $v0,4
syscall

#loads string from user
la $a0,input_str
li $a1,64
li $v0,8
syscall

#initialize Random Number Generator
li $v0,30
syscall

move $a1, $a0

li $v0,40
li $a0,1
syscall

#store address of input_string, to loop through
la $t0,input_str
#store address of input_string, to keep
la $s0,input_str
#initialize length counter to 0
addi $t1,$0,0
#initialize replacements counter to 0
addi $s1,$0,0

#find length of string
length:
      #loads character
      lb $t2,0($t0)
      #exit if character is null (end of string)
      beq $t2,$0,rpl_loop
      #increment counter
      addi $t0,$t0,1
      addi $t1,$t1,1
      j length
```

```
#loop so 4 characters are replaced
rpl_loop:
      move $a0, $s0
      move $a1, $t1
loop:
      beq $s1,4,exit
      jal replace
      addi $s1,$s1,1
      j loop

#replace character
replace:
      #store return address in stack
      addi $sp,$sp,-8
      sw $ra,0($sp)
      sw $s0,4($sp)

      #obtain random number
      jal random_number

      #actually replace character
      addi $t0,$0,0x2A
      add $t1,$a0,$v0
      sb $t0,0($t1)

      #restore old return address
      lw $s0,4($sp)
      lw $ra,0($sp)
      addi $sp,$sp,8
      jr $ra


#generate random number
random_number:
      #store $ra
      addi $sp,$sp,-4
      sw $ra,0($sp)

      move $t0, $a0
rng_loop:
      #get random number
      li $v0,42
      li $a0, 1
      syscall

      add $t1, $t0, $a0
      #check if character is not already a *
      lb $t2,0($t1)
      beq $t2,0x2A,rng_loop

      move $v0, $a0
```

```
      move $a0, $t0

      #restore old return address
      lw $ra,0($sp)
      addi $sp,$sp,4
      jr $ra



#end of program
exit:
      #display message
      la $a0,message
      li $v0,4
      syscall

      #display output
      la $a0,input_str
      li $v0,4
      syscall

      #exit
      li $v0, 10
      syscall
```

Part 2: Recursive Functions

```
.data

prompt:     .asciiz "Enter the sequence index?\n"
message:    .asciiz "The Fibonacci value is:\n"

.text

#displays prompt
la $a0,prompt
li $v0,4
syscall

#gets value from user, moves to general purpose
li $v0,5
syscall

move $a0,$v0
jal fibonacci
move $s0,$v0

#display message
li $v0,4
la $a0,message
syscall
```

```
#display output

move $a0,$s0
li $v0, 1
syscall

#terminate program
li $v0, 10
syscall




#Fibonacci loop
fibonacci:

      bgt $a0,2,continue

      #base case
      addi $v0, $0, 1
      jr $ra


continue:
      #prologue - store
      addi $sp,$sp,-12
      sw $ra,8($sp)
      sw $s0,4($sp)
      sw $s1,0($sp)

      move $s0,$a0
      #n-1
      subi $a0,$s0,1
      jal fibonacci

      move $s1,$v0
      subi $a0,$s0,2
      #n-2
      jal fibonacci
      add $v0,$v0,$s1
      move $a0,$s0

      lw $s1,0($sp)
      lw $s0,4($sp)
      lw $ra,8($sp)
      addi $sp,$sp,12

      #go back
      jr $ra
```