6. In this question, we will write subroutines to access the ADC we have added to our MIPS system. As before, the three pins are mapped to the following addresses:

```
Start   0xFFFF FF00
Done    0xFFFF FF40
DataOut 0xFFFF FF80
```

(a) (5 points) Write a subroutine called `readADC` that will

1. Activate the ADC
2. Wait until the conversion is over
3. Read the Value converted by the ADC and return it to the calling program.

Solution:

```
1 readADC:   addi $t1, $0, 1         # We need '1' in one register
2                                    # Optionally check if 'Done'=0
3            sw   $t1, 0xff00($0)    # Activate ADC by writing '1' to 'Start'
4 waitloop:  lw   $t2, 0xff40($0)    # Read 'Done'
5            beq  $t2, $0, waitloop # if 'Done'= 0 keep waiting
6            lw   $v0, 0xff80($0)    # Read the 'DataOut'
7            sw   $0,  0xff00($0)    # Set 'Start' back to 0
8            jr   $ra                # Return to $ra
```

(b) (5 points) Now write a function `averageADC` that will collect 256 values from the ADC by calling the subroutine you have written above. Average these 256 values and return the result back to a calling subroutine.

Solution:

```
1  averageADC : addi $s0, $0, 0       # this will be our count
2               addi $sp, $sp, -4     # make room on stack
3               sw   $ra, 0($sp)      # push the return address
4               addi $s2, $0, 256     # end count for the loop
5      loop :   jal  readADC          # call subroutine we have written
6               add  $s0, $v0, $s0    # accumulate values in $s0
7               addi $s2, $s2, -1     # reduce count
8               beq  $s2, $0, fin     # jump to loop end
9               j    loop
10       fin :  sra  $v0, $s0, 8      # divide by 256 for averaging
11              lw   $ra, 0($sp)      # read back the return address
12              addi $sp, $sp, 4      # restore stackpointer
13              jr   $ra              # return to $ra
```