CS/COE 0447 Fall 2009
Lab 8: Floating Point Instructions
Solution

```
#Problem 1:
      .data

two:  .float 2.0
three:      .float 3.0


      .text

      li $v0, 6               #Read float
      syscall
      mov.s $f1, $f0          #$f1 holds X

      li $v0, 6               #Read float
      syscall
      mov.s $f2, $f0          #$f2 holds Y

      li $v0, 6               #Read float
      syscall
      mov.s $f3, $f0          #$f3 holds Z

      mul.s $f4, $f1, $f1     #$f4 = X^2

      l.s $f5, two            #$f5 = 2*X*Y
      mul.s $f5, $f5, $f1
      mul.s $f5, $f5, $f2

      l.s $f6, three          #$f6 = 3*Y
      mul.s $f6, $f6, $f2

      sub.s $f5, $f5, $f6     #$f5 =  (2*X*Y - 3*Y) / Z
      div.s $f5, $f5, $f3

      add.s $f12, $f4, $f5    #$f12 = X^2 + (2*X*Y - 3*Y) / Z

      li $v0, 2               #Print float
      syscall

      li $v0, 10              #Exit
      syscall


#Problem 2:
      .data
size: .word 5
a1:   .float 1.3 2.5 6.9 4.4 3.8
a2:   .space 20

zero: .float 0.0
sp:   .asciiz " "
```

```
nl:   .asciiz "\n"


      .text

      lw $t0, size            #Size of arrays
      la $t1, a1              #Start address of a1
      la $t2, a2              #Start address of a2
      li $t3, 0               #Counter
      l.s $f0, zero           #Cumulative sum
start:
      beq $t3, $t0, end       #Branch out of the loop if at end of array
      l.s $f1, 0($t1)         #Load value from array
      add.s $f0, $f0, $f1     #Add it to cumulative sum
      s.s $f0, 0($t2)         #Store it in second array
      addi $t1, $t1, 4        #Increment address to current element of first array
      addi $t2, $t2, 4        #Increment address to current element of second array
      addi $t3, $t3, 1        #Increment counter
      j start                 #Jump back to the top
end:  la $a0, a1              #Print first array
      lw $a1, size
      jal printArray
      lw $a1, size            #Print second array
      la $a0, a2
      jal printArray
      li $v0, 10              #Exit
      syscall

printArray:
      li $t0, 0               #Counter
      move $t1, $a0           #Start address of array to prinf
      la $t2, sp              #Load address of string containing a space
pAStart:
      beq $a1, $t0, pAEnd     #Branch out of the loop if at end of array
      l.s $f12, 0($t1)        #Load value from array
      li $v0, 2               #Print float value
      syscall
      move $a0, $t2           #Print space
      li $v0, 4
      syscall
      addi $t1, $t1, 4        #Increment address to current element of array
      addi $t0, $t0, 1        #Increment counter
      j pAStart               #Jump back to the top
pAEnd:
      la $a0, nl              #Print newline
      li $v0, 4
      syscall
      jr $ra                  #return


#Problem 3:
            .data
size:       .word 6
a1:    .float 3.3 5.5 9.8 10.1 1.3 2.3

plusInf:    .word 0x7f800000
negInf:     .word 0xff800000
```

```
min:  .asciiz "Min: "
max:  .asciiz "\nMax: "


      .text
      lw $t0, size            #Size of arrays
      la $t1, a1              #Start address of a1
      li $t2, 0               #Counter
      l.s $f0, plusInf        #$f0 holds the min value
      l.s $f1, negInf         #$f1 holds the max value
start:
      beq $t2, $t0, end       #Branch out of the loop if at end of array
      l.s $f2, 0($t1)         #Load value from array
      c.lt.s $f2, $f0         #Compare with current minimum
      bc1f endMin             #Skip if not less than
      mov.s $f0, $f2          #Set new minimum
endMin:
      c.lt.s $f1, $f2         #Compare with current maximum
      bc1f endMax             #Skip if not greater than
      mov.s $f1, $f2          #Set new maximum
endMax:
      addi $t1, $t1, 4        #Increment address of current element in array
      addi $t2, $t2, 1        #Increment counter
      j start                 #Jump back to the top
end:  li $v0, 4               #Print "Min: "
      la $a0, min
      syscall
      li $v0, 2               #Print min value
      mov.s $f12, $f0
      syscall
      li $v0, 4               #Print "Max: "
      la $a0, max
      syscall
      li $v0, 2               #Print max value
      mov.s $f12, $f1
      syscall
      li $v0, 10              #Exit
      syscall
```