C335 Homework #2 Solution

Part I

The result for addition: 110000000011
The result for subtraction: 010100101101

Part II

(1)

7ffffffa(hex) = 0111 1111 1111 1111 1111 1111 1111 1010 (bin)
7ffffffa(hex) = 7fffffff(hex) – 5 (dec) = 2,147,483,647 (dec) – 5 (dec)
= 2,147,483,642 (dec)

(2)

11001010111111101111101011001110 (bin) = CAFEFACE(hex)

Part III

(1) -103(dec) = 10011001 (bin) = 0x99
(2) 0xAA = 10101010 (bin) = - 86 (dec)

Part IV

(1)

4096 (dec) = 2^{12} (dec)
= 0000 0000 0000 0000 0001 0000 0000 0000 (bin, 2's complement)

(2)

2047 (dec) = 2048 (dec) – 1
= 2^{11} (dec) – 1
= 0000 0000 0000 0000 1000 0000 0000 (bin) -1
= 0000 0000 0000 0000 0111 1111 1111 1111 (bin)

- 2047 (dec) = (0000 0000 0000 0000 0111 1111 1111)' + 1
= 1111 1111 1111 1111 1000 0000 0000 (bin) + 1
= 1111 1111 1111 1111 1000 0000 0001 (bin, 2's complement)

(3)

X = 1111 1111 1111 1111 1111 1111 0000 0110 (bin, 2's complement)
- X = 0000 0000 0000 0000 0000 0000 1111 1001 (bin)
+ 1
= 0000 0000 0000 0000 0000 0000 1111 1010 (bin)
= 2^8 – 1 – 5 = 256 – 6 = 250 (dec)
So X = - 250 (dec)

(4)

X = 1111 1111 1111 1111 1111 1111 1110 1111 (bin, 2's complement)
-X = 0000 0000 0000 0000 0000 0000 0001 0000 (bin)
+ 1
= 0000 0000 0000 0000 0000 0000 0001 0001 (bin)
= 16 + 1 = 17 (dec)
So X = - 17 (dec)

Part V

```
        add $t0, $zero, $zero       # initialize running sum $t0 = 0
loop:   beq $a1, $zero, finish      # finished when $a1 is 0
        add $t0, $t0, $a0           # compute running sum of $a0
        addi $a1, $a1, -1           # compute this $a1 times
        j loop
finish: addi $t0, $t0, 100          # add 100 to a * b
        add $v0, $t0, $zero         # return a * b + 100
```

The program computes a * b + 100.

Part VI

```
        sll $a2, $a2, 2             # max i= 2500 * 4
        sll $a3, $a3, 2             # max j= 2500 * 4
        add $v0, $zero, $zero       # $v0 = 0
        add $t0, $zero, $zero       # i = 0
outer:  add $t4, $a0, $t0           # $t4 = address of array1[i/4]
        lw $t4, 0($t4)              # $t4 = array 1[i]
        add $t1, $zero, $zero       # j = 0
inner:  add $t3, $a1, $t1           # $t3 = address of array2[j/4]
        lw $t3, 0($t3)              # $t3 = array 2[j/4]
        bne $t3, $t4, skip          # if (array 1[i/4] != array 2[j/4]) skip $v0++
        addi $v0, $v0, 1            # $v0++
skip:   addi $t1, $t1, 4            # j = j+4
        bne $t1, $a3, inner         # loop if j != 2500 * 4
        addi $t0, $t0, 4            # I = i+4
        bne $t0, $a2, outer         # loop if i != 2500 * 4
```

The code determines the number of matching elements between the two arrays and returns this number in register $v0

.

Part VII

```
       addi $v0, $zero, -1    # Initialize to avoid counting zero word
loop:  lw,  $v1, 0($a0)       # Read next word from source
       addi $v0, $v0, 1       # Increment count words copied
       sw   $v1, 0($a1)       # Write to destination
       addi $a0, $a0, 4       # Advance pointer to next source
       addi $a1, $a1, 4       # Advance pointer to next destination
       bne  $v1, $zero, loop  # Loop if word copied != zero
```

Bug 1: Count ($v0) is initialized to zero, not –1 to avoid counting zero word.

Bug 2: Count ($v0) is not incremented.

Bug 3: Loops if word copied is equal to zero rather than not equal.

Bonus Question

(1) If unsigned:
1.1011 (Bin) = 1 x 2^0 + 1 x 2^(-1) + 0 x 2^(-2) + 1 x 2^(-3) + 1 x 2^(-4) = 1.6875 (Dec)
If signed, 2's complement:
1.1011(Bin) = 1 x (-2^0) + 1 x 2^(-1) + 0 x 2^(-2) + 1 x 2^(-3) + 1 x 2^(-4) = -0.3125 (Dec)

(2) 0.1(Bin) = 0.000110011001100...