Name: **Md. Shamiul Islam** **ID:**17301108
Section: 03

Question 1

Translate the following MIPS assembly instructions to machine codes. You need to use your ID to solve this problem.

To find the corresponding register no., follow these instructions:
ID: 17301108
t1 = 5th digit of your ID = 1
t2 = 6th digit of your ID = 1
s1 = 7th digit of your ID = 0
s5 = 8th digit of your ID = 8
s6 = 11 (Fixed)
Do not worry if you get duplicate values.

The first memory address is:
XXXX = (1108) × 4= 4432

Fill up the rest of the values

Instruction	Memory Address	Machine code					
Loop: sll $t2, $s1, 2	4432	0	0	0	1	2	0
add $t2, $t2, $s6	4436	0	1	11	1	0	32
lw $t1, 0($t2)	4440	35	1	1	0		
bne $t1, $s5, Exit	4444	5	8	1	3		
addi $s1, $s1, 2	4448	8	0	0	2		
addi $s1, $s1, -1	4452	8	0	0	-1		
j Loop	4456	2	1108				
Exit: …	4460						

Question 2

Suppose you want to multiply two numbers using the long-multiplication approach in a 10-bit architecture where the product and multiplicand registers are 20-bit and the multiplier register is 10 bit. The multiplicand is X and the multiplier is -(X-1) where X is the sum of all the digits of your BRACU ID (*For example, if ID is 181012141 then X = 19 and -(X-1) = -18*).

Now, multiply X and -(X-1) using the long-multiplication algorithm and show the process in a tabular fashion in the given table.

(You will have to add the necessary number of rows to the provided table to complete the multiplication. The table must be constructed according to the algorithm and the example that was shown in buX and lectures. Finally, kindly mention your ID at the beginning of the solution.)

Answer:

Student ID: 17301108
Multiplicand, X = (1+7+3+0+1+1+0+8) =21 ; Binary form = 00000000000000010101
Multiplier, -(X-1) = -(21-1) = -(20) ; Binary form = 0000010100

Iteration	Multiplier	Multiplicand	Product
0	0000010100	00000000000000010101	00000000000000000000
1	0000010100	00000000000000101010	00000000000000000000
	0000001010	00000000000000101010	00000000000000000000
2	0000001010	00000000000001010100	00000000000000000000
	0000000101	00000000000001010100	00000000000000000000
3	0000000101	00000000000001010100	00000000000001010100
	0000000101	00000000000010101000	00000000000001010100
	0000000010	00000000000010101000	00000000000001010100
4	0000000010	00000000000101010000	00000000000001010100
	0000000001	00000000000101010000	00000000000001010100
5	0000000001	00000000000101010000	00000000000110100100
	0000000001	00000000001010100000	00000000000110100100
	0000000000	00000000001010100000	00000000000110100100
6	0000000000	00000000010101000000	00000000000110100100
	0000000000	00000000010101000000	00000000000110100100

7	0000000000	00000000101010000000	0000000000110100100
	0000000000	00000000101010000000	0000000000110100100
8	0000000000	00000001010100000000	0000000000110100100
	0000000000	00000001010100000000	0000000000110100100
9	0000000000	00000010101000000000	0000000000110100100
	0000000000	00000010101000000000	0000000000110100100
10	0000000000	00000101010000000000	0000000000110100100
	0000000000	00000101010000000000	0000000000110100100

Product is : 0000000000110100100 ; Decimal form: 420

2's complement of product : 1111111111001011011 +1 = 1111111111001011100

Question 3

Write down the MIPS code for the following C code:

```
Int task02(int x, int y) {
        x = x + y;
        if (x < M) {
                z = x - y;
        else {
                z = y - x;
        }
        return z;
}
```

Here M is a constant integer value, which is the sum of all the digits of your ID. Use $s0 for z.

Answer:

M = 1+7+3+0+1+1+0+8 = 21
here ,
x → $a0
y → $a1

```
task02:
        addi $sp, $sp, -12
        sw $ra, 8($sp)
        add $a0, $a0, $a1
        sw $a1, 4($sp)
        sw $a0, 0($sp)
        slti $t0, $a0, 21
        bne $t0, $zero, if
        sub $s0, $a1, $a0
        add $v0, $s0, $zero
        addi $sp, $sp, 12
        jr $ra

        if:
        lw $ra, 8($sp)
        lw $a1, 4($sp)
        lw $a0, 0($sp)
        sub $s0, $a0, $a1
        add $v0, $s0, $zero
        addi $sp, $sp, 12
        jr $ra
```