Name: Ariful Hoque **ID**: 17301107

Section: 02

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 : 17301107
t1 = 1
t2 = 1
s1 = 0
s5 = 7
s6 = 11 (Fixed)
Do not worry if you get duplicate values.

The first memory address is:
XXXX = (1107) × *4* = 4428

Fill up the rest of the values

Instruction	Memory Address	Machine code					
`Loop: sll $t2, $s1, 2`	4428	0	0	0	1	2	0
`add $t2, $t2, $s6`	4432	0	1	11	1	0	32
`lw $t1, 0($t2)`	4436	35	1	1	0		
`bne $t1, $s5, Exit`	4440	5	7	1	3		
`addi $s1, $s1, 2`	4444	8	0	0	2		
`addi $s1, $s1, -1`	4448	8	0	0	-1		
`j Loop`	4452	2	1107				
`Exit: …`	4456						

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 bits. 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.

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

Answer:

Student ID: 17301107

Multiplicand, X = 1+7+3+0+1+1+0+7 = $(20)_{10}$ = (binary of 20 :0000000000 0000010100$)_2$

Multiplier, -(X-1) = -(20-1) = -19 = (binary of 19: 0000010011$)_2$

Iteration	Multiplier	Multiplicand	Product
0	0000010011	00000000000000010100	00000000000000000000
	0000010011	00000000000000010100	00000000000000010100
1	0000010011	00000000000000101000	00000000000000010100
	0000001001	00000000000000101000	00000000000000010100
	0000001001	00000000000000101000	00000000000000111100
2	0000001001	00000000000001010000	00000000000000111100
	0000000100	00000000000001010000	00000000000000111100
3	0000000100	00000000000010100000	00000000000000111100
	0000000010	00000000000010100000	00000000000000111100
4	0000000010	00000000000101000000	00000000000000111100
	0000000001	00000000000101000000	00000000000000111100
5	0000000001	00000000000101000000	00000000000101111100

	0000000001	00000000001010000000	00000000000101111100
	0000000000	00000000001010000000	00000000000101111100
6	0000000000	00000000010100000000	00000000000101111100
	0000000000	00000000010100000000	00000000000101111100
7	0000000000	00000000101000000000	00000000000101111100
	0000000000	00000000101000000000	00000000000101111100
8	0000000000	00000001010000000000	00000000000101111100
	0000000000	00000001010000000000	00000000000101111100
9	0000000000	00000010100000000000	00000000000101111100
	0000000000	00000010100000000000	00000000000101111100
10	0000000000	00000101000000000000	00000000000101111100
	0000000000	00000101000000000000	00000000000101111100

Product → **0000000000 0101111100 (decimal: 380)**
2's complement of product -> 1111111111 1010000011+1 -> 1111111111 1010000100

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+7 = 20.

x -> $a0

y -> $a1

```
jal task02
task02:
     addi $sp, $sp, -4
     sw $s0,0($sp)
     add $a0, $a0, $a1
     slti $t0, $a0,20
     bne $t0, $zero,if
     sub $s0, $a1, $a0
     add $v0, $s0, $zero
     lw $s0,0($sp)
     addi $sp, $sp,4
     jr $ra

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