ICS 233: Computer Architecture & Assembly Language
Fall Semester 2021 (211) – Section 01
Quiz 3

Name: *ID#*

Q1. (4 points) Implement the following high-level statement assuming that variables *a*, *b*, and *c* are signed integer values and are loaded to registers *$t0*, *$t1*, and *$t2* respectively:

```
if ((a < 10 || a >= 100) && a > b) {c = 1;}
```

Solution:

```
    blt $t0, 10, L1        # if a < 10, then check if a > b
    blt $t0, 100, skip     # if a >= 10 and a < 100, skip if
L1: ble $t0, $t1, skip     # if a <= b, skip if
    li $t2, 1
skip: . . .
```

Q2. (3 points) Assuming that variable Array is defined as shown below:

```
Array:   .byte -1, 2, -3, 4
```

After executing the following sequence of instructions, the content of the registers $t1 and *$t2* is $t1=0x_FFFF FFFD_, $t2=0x_0000 04FD_. You can assume little endian byte ordering.

```
    la $t0, Array
    lb $t1, 2($t0)
    lhu $t2, 2($t0)
```

Q3. (3 points) Given a two-dimensional **array[10][20]** of 10 rows and 20 columns defined as follows:

```
array:   .half   0:200
```

write the minimal assembly code to load element **array[5][10]** to register $t1. Assume the value stored at the array are unsigned numbers.

solution:

```
    la $t0, array
    addui   $t0, $t0, 110    # 5 x columns + 10 = 110
    sll $t0, $t0, 1
    lhu $t1, 0($t0)
```

ICS 233: Computer Architecture & Assembly Language
Fall Semester 2021 (211) – Section 02
Quiz 3

Name: *ID#*

Q1. (4 points) Implement the following high-level statement assuming that variables a, b, and c are signed integer values and are loaded to registers $t0, $t1, and $t2 respectively:

```
        if ((a > 10 && a <= 100) || a > b) {c = 1;}
```

```
Solution:
    ble $t0, 10, L1       # if a <= 10, then check if a > b
    ble $t0, 100, if      # if a <= 10 , execute if block
L1: ble $t0, $t1, skip    # if a <= b, skip if
if: li $t2, 1
skip: . . .
```

Q2. (3 points) Assuming that variable Array is defined as shown below:

```
Array:    .byte -1, 2, -3, 4
```

After executing the following sequence of instructions, the content of the registers $t1 and *$t2* is $t1=0x_0000 00FD_, $t2=0x_0000 04FD_. You can assume little endian byte ordering.

```
    la $t0, Array
    lbu $t1, 2($t0)
    lh $t2, 2($t0)
```

Q3. (3 points) Given a two-dimensional **array[20][10]** of 20 rows and 10 columns defined as follows:

```
array:    .word    0:100
```

write the minimal assembly code to load element **array[15][5]** to register $t1 Assume the value stored at the array are unsigned integers.

solution:

```
    la $t0, array
    addui    $t0, $t0, 155    # 15 x columns + 5 = 110
    sll  $t0, $t0, 2
    lw   $t1, 0($t0)
```