CS220: Computer Organization
Quiz#3

Name:
Roll No.:

General instructions: In all the questions, you will assume 32-bit big-endian MIPS ISA.

1. Consider translating the following for loop into MIPS. Assume that `i` and `N` allocated in registers $t0 and $t1, respectively.

```
for (i=0; i<N; i++) { loop body }
```

The skeleton of the MIPS translation is shown below. The `bne` instruction starts at address 0x00603000. What is the minimum possible address of the first instruction of the loop situated at the label `start`? Express your answer in hexadecimal. **(2 points)**

```
       ...
start: ...
       loop body
       ...
       slt $t2, $t0, $t1
       bne $t2, $0, start
```

Solution: The negative value with largest magnitude for the PC-relative offset of the `bne` instruction is 0x8000. After sign-extension and shifting left by two bit positions, we get 0xfffe0000. The minimum value of the label `start` is obtained by adding this to 0x00603000. Therefore, the minimum value of the label `start` is 0x005e3000.

Grading policy: No partial marks.

2. Consider the following MIPS instruction sequence. What is the final hexadecimal value in $t0? **(2 points)**

```
lui  $t0, 0x62
addi $t0, $t0, 0xaabc
sra  $t0, $t0, 0x4
```

Solution: After the `lui` instruction, $t0 has 0x00620000. The immediate operand of the `addi` instruction is sign-extended and added to $t0. After the `addi` instruction, $t0 has 0x0061aabc. When this is shifted to right by four bit positions, we get 0x00061aab.

Grading policy: One mark if the contents of $t0 are correct after the `addi` instruction. One more mark for correct execution of the shift instruction.

3. Consider the following MIPS instruction sequence. Assume that initially $t0 contains 0x10000000 and $t1 contains 0x10000004. Initially, the word stored at address 0x10000000 is 0x12fe43ba and the word stored at address 0x10000004 is 0xab34ef21. What is the final hexadecimal value of the word stored at address 0x10000004? **(3 points)**

```
lb $t0, 1($t0)
sh $t0, 0($t1)
```

Solution: Since MIPS is big-endian, the `lb` instruction loads 0xfe into $t0 and sign-extends it. So, $t0 has 0xfffffffe after `lb` instruction. The `sh` instruction stores the least significant half-word from $t0 into addresses 0x10000004 and 0x10000005. So, the final value of the word at address 0x10000004 is 0xfffeef21.

Grading policy: No partial marks.

4. Consider translating the following C statement where the value of `label` is 0x0 and `label1` is 2^{28} instructions away. This information is available at the time of compilation of the statement. Show the MIPS translation of this C statement using minimum number of instructions. **(3 points)**

```
label: goto label1
```

Solution: The value of `label1` is 2^{30} i.e., 0x40000000. The instruction sequence is shown below.

```
lui $at, 0x4000
jr $at
```

Grading policy: Any extra instruction or wrong immediate operand in the `lui` instruction will have a penalty of one mark, provided everything else is correct. If the instruction sequence is completely different, no partial marks.