CS220 Quiz#3

General instructions: Please write brief explanation for your answers. If you submit multiple times, your last submission will be used for grading. Please provide an email address below where your responses can be sent.

Email address *

mainakc@cse.iitk.ac.in

Your name *

Mainak Chaudhuri

Your roll number *

0

Q1. Write the branch instruction names of the MIPS ISA that will be generated by a reasonably optimized compiler from this C statement: if ((x != y) || (x <= 0)). [1 point]

bne, blez

Q2. Write the branch instruction names of the MIPS ISA that will be generated by a reasonably optimized compiler from this C statement: if ((x != y) && (x < 0)). [1 point]

beq, bgez

Q3. Consider a function f written using the C language. The function f calls another function g having twelve arguments all of type integer. The function f can allocate all its local variables in registers without spilling. The structure of f is as follows: {... return g (a, b, c, d, e, ...);}. How much stack space in bytes should be allocated to f when compiling for 32-bit MIPS? [1 point]

f needs to save $ra on stack before calling g. It also needs to put eight arguments of g on stack. Therefore, f needs 36 bytes of stack space.

Q4. Consider the following sequence of 32-bit MIPS instructions separated by semi-colons: [addi $t0, $0, 0xf2; sll $t0, $t0, 0x18; addi $t1, $0, 0x2; srav $t0, $t0, $t1]. What is the final hexadecimal value in $t0? [1 point]

```
addi $t0, $0, 0xf2     // $t0 <-- 0xf2
sll    $t0, $t0, 0x18   // $t0 <-- 0xf2000000
addi  $t1, $0, 0x2      // $t1 <-- 0x2
srav  $t0, $t0, $t1      // $t0 <-- 0xfc800000
```

Q5. Consider the following segment of C code: [for(i=0;i<20;i++) { if (i%2==0) { // Some non-branch statements } else if (i%3==0) { // Some non-branch statements } }]. This code is translated to 32-bit MIPS such that the translated code has the minimum number of branch/jump instructions. When the translated code is executed, calculate how many forward branches/jumps and how many backward branches/jumps are executed. [1+1 points]

Both if statements translate to forward branches. The branch for the for loop is a backward branch. The branch for if (i%2 == 0) is executed for all i, but the branch for if (i%3 == 0) is executed for only odd i. Thus, 30 forward branches and 20 backward branches are executed.