Branches and Jumps

#1. True/False

[F] Jump commands (in the MIPS architecture) are always conditional.

[T] Branch commands (in the MIPS architecture) are always conditional.

[T] Jump commands (in the MIPS architecture) are always unconditional.

[F] Branch commands (in the MIPS architecture) are always unconditional.

[T] Jumps are always absolute.

[T] Branches are always relative.

#2.   Our MIPS architecture has only "Branch on Equal" (BEQ) and "Branch on Not Equal" (BNE) commands. Show how to use the "Set on Less Than" (SLT) and BEQ or BNE to make the equivalent of BLT, BGT, BLE, and BGE.

Example: blt $10, $8, label

   slt $5, $10, $8   - sets $5 to true (1) if $10 < $8, to false (0) otherwise.
   bne $5, $0, label

#3. Give two reasons why there are only BEQ and BNE commands:

   a. MIPS is a Reduced Instruction Set Computer (RISC).
   b. Other branch instructions would reqirire four more opcodes, which might cause the total
       number of opcodes to exceed the 64 availablle with only 6 bits used in the code word.

#4. What is the range of a BNE or BEQ instruction?

   No. Bits = 32 – 6 (op code) – 5 (Reg. Addr.) + 2 (added by <<2) = 23
   Range = +/- 2^{22} = +/- 4M   (note: you can use M to mean 2^{20}, approx. 1 million)

#5. What is the range of a "j" jump instruction?

   No. Bits = 32 – 6 (op code) + 2 (added by <<2) = 28
   Range = 2^{28} = 256M

#6. What is the range of a "jr" jump instruction?

   No. Bits = 32 from register whose address is in the instruction (no <<2 needed)
   Range = $2^{32}2$ = +/- 4G (note: you can use G to mean 2^{30}, approx. 1 billion)

#7   What would be the relative address encoded in the following bne instruction (decimal addresses are used rather than hex as in slides)?

```
Memory     Assembly
Address    Instruction

1204  labelx:     add $5, $5, $6
 . . .
1506  bne $7, labelx

Answer: the rightmost 21 bits would be (in decimal)
        [1204 - (1506 +4)]/4 = -306 (words, relative)
```

#8   What would be the address encoded in the following "j" instruction (decimal addresses are used rather than hex as in slides)?

```
Memory     Assembly
Address    Instruction

1204  labelx:     add $5, $5, $6
 . . .
1506  j labelx

Answer: the rightmost 26 bits would be (in decimal)
        [1204 /4 = 301 (words, absolute)
```

#9   What would be the address needed in $11 for the following "jr" instruction to work properly (decimal addresses are used rather than hex as in slides)?

```
Memory     Assembly
Address    Instruction

1204  labelx:     add $5, $5, $6
 . . .
1506  jr $11, labelx

Answer: the 32 bits in $11 would be (in decimal)
        1204  (bytes, absolute)
```

```
#10.  What is the difference between the "jal" and the "j"
command?

Answer: The "jal" command is used for jumping to a subroutine or
procedure, so the value PC+4 (the next instruction) is stored in
$ra ($31).

#11.  What command is used at the end of a subroutine or
procedure?   Answer "jr $ra".
```