Ruth Adams
ELEC 5200
9-14-21

23/26

Homework 3

✓ 1.) LUI x11, 0x87654 // upper 20 bits w/ given
ADDI x11, x11, 0x321 // lower 12 bits w/ given

✓ 2.) a.) krs1 is 6 lines behind its call, so PC-24 will be our constant, or [...1101000] (20 bits)

✓ b.) krs2 is 2 lines ahead of its call, so PC+8 will be our constant, or [...001000] (20 bits)

✓ c.) imm[12|10:5] | rs2 | rs1 | f3 | imm[4:1|11] | opcode *bge line
0 (7) (5) (5) (3) (5) (7)
0000000 01011 00101 101 01000 1100011

+ d.) *jal line
imm[20|10:1|11|19:12] | rd | opcode
(20) (5) (7)
drop lowest 1111101000111111100000 1101111
imm bit

-2 3.) Loop: lw x3, 0(x4) ← init x6
sw x3, 0(x5)
beq x3, x0, Exit // b/c we copy but don't increment if @ terminating word
addi x6, x6, 1 // word has been checked + doesn't meet condition
slli x4, x4, 2 > 4 not 2
slli x5, x5, 2
jal x0, loop // start loop again
Exit:

4.) a.) -2 addi sp, sp, ~~-8~~ -4 // room for 1 reg
sw ~~x6~~ x25, 0(sp) // save y in sp
slli x5, x10, 1 // 2x x5 is a saved register
add x25, x11, x5 // y=2x+b
mv x12, x25 // move y to saved reg. save to x10
lw x25, 0(sp)
addi sp, sp, 4
jalr x0, 0(x1) // return

4b.)
```
int foo(int *x, int *b) {
    int y;
    y = 2*(*x) + (*b);
    return y;
}
```

4c.)
```
foo: addi sp, sp, -4 //space for y
     sw   x25, 0(sp)
     lw   x28, 0(x10)  //load x
     slli x30, x28, 1  //x30 = 2x
     lw   x29, 0(x11)  //load b
     add  x25, x30, x29  //y = 2x+b
     sw   x28, 0(x10) }  unnecessary
     sw   x29, 0(x11) }
     mv   x12, x25
     lw   x25, 0(sp)
     addi sp, sp, 4
     jalr x0, 0(ra)
```