Alexander Miller
Homework 5

1h.
lb $s3, 100($a0)

opcode [6 bits]: 32 = 100000
rs [5 bits]: $a0 = 4 = 00100
rt [5 bits]: $s3 = 19 = 10011
imm [16 bits]: 100 = 0000000001100100

Binary Encoding: 1000_0000_1001_0011_0000_0000_0110_0100
Hex Encoding: 0x80930064

1i.
addi $sp, $sp, -32

opcode [6 bits]: 8 = 001000
rs [5 bits]: sp = 29 = 11101
rt [5 bits]: sp = 29 = 11101
imm [16 bits]: -32 = 0100000 -> 1011111 -> 1100000 -> 1111111111100000

Binary Encoding: 0010_0011_1011_1101_1111_1111_1110_0000
Hex Encoding: 0x23bdffe0

2h.
Hex Encoding: A5583BC9
Binary Encoding: 1010_0101_0101_1000_0011_1011_1100_1001

opcode: 101001 = 32+8+1 = 41 = sh
I format: 101001_01010_11000_0011101111001001
rs: 01010 = 2+8 = 10 = $t2
rt: 11000 = 8+16 = 24 = $t8
imm: 0011101111001001 = 2^0+2^3+2^6+2^7+2^8+2^9+2^11+2^12+2^13
so imm = 1+8+64+128+256+512+2048+4096+8192 = 15305

Command:
sh $t8, 15305

2i.
Hex Encoding: 020B0823
Binary Encoding: 0000_0010_0000_1011_0000_1000_0010_0011

opcode[6 bits]: 000000

R-format: 000000_10000_01011_00001_00000_100011
rs [5 bits]: 10000 = 16 = $s0
rt [5 bits]: 01011 = 11 = $t3
rd [5 bits]: 00001 = 1 = $at
shamt [5 bits]: 00000 = 0
funct [6 bits]: 100011 = 35 = subu

Command:
subu $at, $s0, $t3

3e.
* $t0 in instruction2 is dependent on instruction1
* $t1 in instruction4 is dependent on instruction2
* $t2 in instruction4 is dependent on instruction3

3f.
* $s2 in instruction3 is dependent on instruction1
* $s5 in instruction3 is dependent on instruction2
* $s6 in instruction4 is dependent on instruction3

4d.

```
int problemPartD() {
        int i = 0;
        while (1) {
                int returnVal = getAValue();
                if (returnVal == 0) {
                        break;
                }
                i = i + returnVal;
        }
        return i;
}
```