HW1
Robert Loquinario
CMPE 12

1. No, a higher-level programming language cannot instruct a computer to do more than a lower-level programming language, but is easier to use.

2. It's increasing harder to improve accuracy of analog computers.

3. Three characteristics of algorithms:
    1. Definiteness: the algorithm has precise steps.
    2. Effective Computability: each of the steps can be carried out by a computer.
    3. Finiteness: The algorithm will finish.

4. One advantage higher-level languages have compared to lower-languages is that they are easier to use. A disadvantage would be performance of the program, as lower level languages have the ability to optimize better than compliers.

5. Three things specified by an ISA (instruction set architecture) are instructions, data types, and addressing modes.

6. Microarchitecture specifies how circuits are put together to make a computer. Instruction Set Architecture (ISA) provides an interface which specifies what sort of instructions a computer supporting this interface can perform. A microarchitecture is a detailed organization of implementing an ISA.

Base	0	1	2	3	4	5	6	7
2	1	2	4	8	16	32	64	128
3	1	3	9	27	81	243	729	2187
4	1	4	16	64	256	1024		
5	1	5	25	125	625			
6	1	6	36	216				

7. numbers to unsigned binary
    a. 26 : 16 + 8 + 2 = 00011010_2
    b. 49 : 32 + 16 + 1 = 00110001_2
    c. 255 : 128+64+32+16+8+4+2+1= 11111111_2
    d. 129 = 128+1 = 10000001_2

8. binary to base 10 and hexadecimal
    a. 00101010_2:
        - $32+8+2 = 42_{10}$

- 0010=2 , 1010=10=A : 0x2A

b. 00111111_2:
- $32+16+8+4+2+1 = 63_{10}$
- 0011_2=3, 1111_2=15=F : 0x3F

c. 10000000_2:
- 128_{10}
- 1000_2 = 8, 0000_2=0 : 0x80

d. 11101001_2:
- $128+64+32+8+1=233_{10}$
- 1110_2=14 =E, 1001_2=9 : 0xE9

e. 00001001_2:
- $8+1=9_{10}$
- 0000_2=0, 1001_2=9 : 0x09

9. convert to following bases

a. (3): $81+9+9+1 = 10201_3$

b. (4): $64+16+16+4 = 1210_4$

c. (5): $25+25+25+25 = 400_5$

d. (6): $36+36+6+6+6+6+1+1+1+1 = 244_6$

10. change to base 10

a. 210_3 : $9 +9+3=21_{10}$

b. 321_4 :$16+16+16+4+4+1=57_{10}$

c. 432_5 : $25+25+25+25+5+5+5+1+1=117_{10}$

11. Convert following numbers to 8-bit 2's complement and perform operation in binary

a. -6 + 20 : 11111010_2 [inverted 6: 00000110_2 -> 11111001_2 add 1]
+ 00010100_2 [16+4]
= 00001110_2 =8+4+2= 14

b. 67 + 30 : 01000011_2 [64+2+1]
+ 00011110_2 [16+8+4+2]
= 01100001_2=64+32+1=97

c. 42 - 20 : 00101010_2 [32+8+2 = 42]
+ 11101100_2 [inverted 20: 00010010_2 -> 11101101_2 add 1]
= 00010110_2=16+4+2=22

d.-44 - 23 : 11010100_2 [inverted 44: 00101100_2 -> 11010011_2 add 1]
+ 11101001_2 [inverted 23: 00010111_2 -> 11101000_2 add 1]
= 10111101_2=- 128+32+16+8+4+1=-67

e. 26 - 26 : 00011010_2 [16+8+2]
+ 11100110_2 [inverted 26 ^ -> 11100101]
= 00000000_2=0