CS 250: Homework 1

Problem 1:

a. i. $0111_B = (1*2^0) + (1*2^1) + (1*2^2) = 7_{16}$

$1000_B = (1*2^3) = 8_{16}$

$1100_B = (1*2^2) + (1*2^3) = 12_D = C_{16}$

$0010_B = (1*2^1) = 2_{16}$

Thus, $0111\ 1000\ 1100\ 0010_B$ = **$78C2_{16}$**

$65/16 = 4R1$ so 1_{16}

$4/16 = 0R4$ so 4_{16}

Thus, 65_D = **41_{16}**

ii. The lowest number one can represent using 16-bit two's complement is:

$1000\ 0000\ 0000\ 0000_B$ which is $-(0111\ 1111\ 1111\ 1111 + 1)$ = **-32768**

The highest number one can represent using 16-bit two's complement is:

$0111\ 1111\ 1111\ 1111_B$ which is **32767**

iii. -17_D:

$17/2 = 8R1$

$8/2 = 4R0$

$4/2 = 2R0$

$2/2 = 1R0$

$1/2 = 0R1$ so $17_D = 0001\ 0001$ so $-17_D = 1110\ 1110 + 1$ = **$1110\ 1111_B$**

$+33_D$:

$33/2 = 16R1$ $16/2 = 8R0$ $8/2 = 4R0$ $4/2 = 2R0$ $2/2 = 1R0$ $1/2 = 0R1$

So 33_D = **$0010\ 0001_B$**

iv. $+15.5_D$:

15/2 = 7R1 7/2 = 3R1 3/2 = 1R1 1/2 = 0R1

0.5 * 2 = 1.0 so $15.5_D = 1111.1_2$

1111.1_2 * 2^0 = 1.1111_2 * 2^3 so Mantissa is 1111

Exponent is 3+127=130=10000010_2

Sign bit is 0. So $+15.5_D$ is **0 1000 0010 111 1000 0000 0000 0000 0000**.

-1.875_D:

0.875 * 2 = 1.75 0.75 * 2 = 1.5 0.5 * 2 = 1 so $1.875_D = 1.111_2$

Mantissa is 111 and exponent is 0+127=127=011111111_2

Sign bit is 1 so it is **1 0111 1111 111 0000 0000 0000 0000 0000**

v. 0x40C40000:

0000_{16} = 0000 0000 0000 0000_2, $4_{16} = 0100_2$, $C_{16} = 12_D = 1100_B$ so it is:

0 1000 0001 100 0100 0000 0000 0000 0000

Sign = 0 (so positive), exponent = (128+1) – 127 = 2, Mantissa = 10001

1.10001*2^2 = 110.001 = 6.125

vi. In Decimal, E = 069, C = 067, E= 069, 2= 050, 5 = 053, 0 = 048, space = 032,

i = 105, s = 115, space = 032, t = 116, h = 104, e = 101, space = 032, b = 098,

e = 101, s = 115, t = 116, ! = 033

So it is: 069 067 069 050 053 048 032 105 115 032 116 104 101 032 098 101 115

116 033

b. fdb
c. gbfb

Problem 2:

a. i. a lives in stack because it is a local variable
   ii. a_ptr lives in data because it is a global variable
   iii. b_ptr lives in heap because more memory is needed to be allocated for it using malloc
   iv. c lives in stack because it is the return address of the instructions being called in foo

b. main returns 0 because c is equal to 12.0
c. Foo returns 12.0 because *y is 5.0 and *z is 7.0 and *x is 2.0 in the foo function so *y+*z is greater than *x*5