1. Convert the following numbers from base 10 to 2 and then to 16:

15

15 / 2 = 7, 15 mod 2 = 1

7 / 2 = 3, 7 mod 2 = 1

3 / 2 = 1, 3 mod 2 = 1

1 / 2 = 0,1 mod 2 = 1

=> 15(10)=1111(2)=F(16)(rapid conversions)

2. Convert the following numbers from base 10 to 16 and then to 2:

11

11(10)=B(16)=1011(2)(rapid conversion)

3. Convert the following numbers from base 2 to 16:

10001010

10001010(2) = 8A(16)(rapid conversion)

4. Convert the following numbers from base 16 to 2:

2F8 = 1011111000(rapid conversion)

5. Compute the following expressions directly in base 2 (without converting to base 10):

1010-1

1010 -  
 1  
------  
1001  
1010-1=1001

6. Compute the following expressions directly in base 16 (without converting to base 10):

10+A  
10+A=1A

7. Check, using at least two of the complementary code rules, if:

(4BA1)16  and (5C93)16 are complementary in a location of 2 bytes  
1st rule: inverted bits + 1  
4BA1(16) = 0100101110100001(2)  
-4BA1(16) = 1011010001011110 + 1 = 1011010001011111(2) = B45F(16) != 5C93(16) => not complementary

2nd rule:Subtract the hexadecimal content of the location from 100..00, where the number of zeros is equal to the number of hexadecimal digits of the location that needs to be complemented.

-4BA1(16) = 10000(16) – 4BA1(16)

10000 -

4BA1

--------

B45F

B45F != 5C93 => not complementary

8. Write the 8-bits unsigned representation for the following numbers:

230  
230 / 2 = 115, 230 mod 2 = 0

115 / 2 = 57, 115 mod 2 = 1

57 / 2 = 28, 57 mod 2 = 1

28 / 2 = 14,28 mod 2 = 0

14 / 2 = 7, 14 mod 2 = 0

7 / 2 = 3, 7 mod 2 = 1

3 / 2 = 1,3 mod 2 = 1

1 / 2 = 0,1 mod 2 = 1

=>230(10)=11100110(2)(corresponds to unsigned representation)

9. Write the 16-bits signed representation for the following numbers:

-6

6 / 2 = 3,6 mod 2 = 0

3 / 2 = 1,3 mod 2 = 1

1 / 2 = 0,1 mod 2 = 1

=>6(10)=110(2)=00000000000000000000000000000110(16 bits)

-6(10) = ~(00000000000000000000000000000110) + 1 =

11111111111111111111111111111001 + 1 =  
 11111111111111111111111111111010