1. Write a function that inverts the bits of an unsigned char x and stores answer in y.

Your answer should print out the result in binary form

Your output should be like this:

x = 10101011 (binary)

x inverted = 01010100 (binary)

2. Write a function that rotates (**NOT shifts**) to the right by n bit positions the bits of an unsigned char x.ie no bits are lost in this process.

Your answer should print out the result in binary form

Your output should be like this:

x = 10100111 (binary)

x rotated by 3 = 11110100 (binary)

3. Write a function setbits(x,p,n,y) that returns x with the n bits that begin at position p set to the rightmost n bits of an unsigned char variable y (leaving other bits unchanged).

E.g. if *x* = 10101010 (170 decimal) and *y* = 10100111 (167 decimal) and *n* = 3 and *p* = 6 say then you need to strip off 3 bits of y (111) and put them in x at position 10*xxx*010 to get answer 10111010.

Your answer should print out the result in binary form

Your output should be like this:

x = 10101010 (binary)

y = 10100111 (binary)

setbits n = 3, p = 6 gives x = 10111010 (binary)

4. Write a function setbits(w,i,j,value) that sets the bits from i to j (inclusive) to given value (0 or 1). Assume w is an unsigned int (32-bits). Check the ranges to make sure i and j falls within the range 0-31