Problem 1:
Convert the 2-digit two's complement hexadecimal integer 0xe6 to decimal. (20 points)

\[ 0xe6 = 1110 \ 0110 = -128 + 64 + 32 + 6 = -26 \]

Problem 2:
Convert the decimal integer 38 to an 8-bit two's complement binary integer. (20 points)

\[ 38 = 2^5 + 2^2 + 2^1 = 100110 \]

Problem 3:
You're given two 4-digit, 2's complement hexadecimal numbers \( X = 0x92c3 \) and \( Y = 0xc1b4 \). Compute \( X-Y \). Remember to indicate overflow if it occurs. Show all intermediate steps clearly. (30 points)

\[ X - Y = 92c3 + 3e4c = d10f \]
Problem 4:

Consider the C++ function reverseList(). reverseList() has two arguments, in order:

- list[] an integer array (input/output)
- count an integer; number of elements in list[] (input)

reverseList() reverses the order of the elements of the array list[]. Write the code for the function reverseList(). Show the prototype for reverseList() clearly.

Write efficient code; obviously inefficient code will be penalized. You are not allowed to call any library function that will reverse the list for you.

(30 points)

```c
void reverseList(int list[], int count)
{
    int i;

    for (i = 0; i < count / 2; i++) {
        int temp = list[i];
        list[i] = list[count - i - 1];
        list[count - i - 1] = temp;
    }
}
```