Problem 1:
Convert the decimal integer $-41$ to 2-digit two's complement hexadecimal. (20 points)

ANSWER: 0xd7

$41 = 32 + 9 = 0010\ 1001$

$-41 = 1101\ 0111 = \text{d7}$

Problem 2:
Convert the 8-bit two's complement binary integer 0101 0011 to decimal. (20 points)

ANSWER: 83

$0101\ 0011 = 64 + 16 + 3 = 83$

Problem 3:
You're given two 4-digit, 2's complement hexadecimal numbers $X = 0\text{343c}$ and $Y = 0\text{5ecd}$. Compute $X - Y$. Remember to indicate overflow if it occurs. Show all intermediate steps clearly. (30 points)

ANSWER: 0xd56f

$-Y = ffff - 5ecd + 1 = \text{a133}$

$X + (-Y) = 343c + \text{a133} = \text{d56f}$
Problem 4:

Consider the C++ function findMin. findMin has three arguments, in order:

- x[] an integer array (input)
- count an integer; number of elements in x[] (input)
- minIndex an integer; index of smallest element in x[] (output)

Write the code for the function findMin(), which finds the smallest element in x[], places the index of the smallest element in minIndex, and returns the value of the smallest element (i.e., if x[8] = -1 is the smallest element, findMin() places 8 in minIndex and returns -1). Show the prototype for findMin() clearly.

Write efficient code; obviously inefficient code will be penalized. (30 points)

```cpp
int findMin(int x[], int count, int& minIndex)
{
    int i, mini, minTemp;

    mini = 0;
    minTemp = x[0];

    for (i=1; i<count; i++) {
        if (x[i] < minTemp) {
            mini = i;
            minTemp = x[i];
        }
    }
    minIndex = mini;
    return minTemp;
}
```