

3. Recall the "grade-school" algorithm for subtraction of two unsigned decimal integers:

```
update = minu;
for (digit = 0; digit < max_digits; digit++) {
    if (update[digit] < subt[digit]) {
        update[digit] += 10;
        i = digit + 1;
        while (update[i] == 0) {
            update[i] = 9;
            i++;
        }
        update[i]--;
    }
    diff[digit] = update[digit] - subt[digit];
}
```

→

```
update = minu;
for (digit = 0; digit < max_digits; digit++) {

    if (update[digit] < subt[digit]) {
        update[digit] += 10;
        i = digit + 1;

        while (update[i] == 0) {
            update[i] = 9;
            i++;
        }
        if (i > max_digits) { //overflow
            printf("minuend is less than the subtrahend");
            exit();
        }
        update[i]--;
    }
    diff[digit] = update[digit] - subt[digit];
}
```

3.1 What is 5ED4–07A4 when these values represent unsigned 16-bit hexadecimal numbers? The result should be written in hexadecimal. Show your work.

```
5ED4(16-bit unsigned) - 07A4(16-bit unsigned)
= 0101 1110 1101 0100 - 0000 0111 1010 0100
= 0101 0111 0011 0000
= 5730(hex)
```

3.2. What is 5ED4–07A4 when these values represent signed 16-bit hexadecimal numbers stored in sign-magnitude format? The result should be written in hexadecimal. Show your work

5ED4(16-bit signed) - 07A4(16-bit signed)

In sign magnitude format, the most significant bit is a sign bit. In order to have a sign bit set in hex, the leading 'character' must be 8-F. In none of the given numbers is this the case. This makes the answers to this problem, the exact same as the one above. So it is 5730(hex)

3.6. Assume 185 and 122 are unsigned 8-bit decimal integers. Calculate 185–122. Is there overflow, underflow, or neither?

185(8-bit unsigned) = 10111001(+)

122(8-bit unsigned) = 01111010(+)

185 - 122

= 1011 1001

- 0111 1010

= 0011 1111(+)

There is no overflow or underflow.

3.7 Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185+122. Is there overflow, underflow, or neither?

185(8-bit signed) = 10111001(-)

two complement of 185 = $\sim 011\ 1001 + 1 = 0100\ 0111$

122(8-bit signed) = 01111010 (+)

185 + 122

= 0100 0111

+ 0111 1010

= 1100 0001

There is no overflow or underflow.

3.8 Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185–122. Is there overflow, underflow, or neither?

185(8-bit signed) = 1011 1001 (-)

122(8-bit signed) = 0111 1010(+)

two complement of 122 = 1000 0110

185 -122

= 185 + (-122)

= 1011 1001

+ 1000 0110

= 0011 1111

There is overflow.