PROGRAMMING PROBLEM:

We are interested in creating a C program that embodies the logic of a simple "Codemaker vs. Codebreaker" game (based on the game called MasterMind – see http://www.archimedes-lab.org/mastermind.html).

For this problem, instead of four colored spheres, the Codemaker generates a positive integer value N which is made up of 3 unique digits. Examples of valid values are 123, 854, 609. Examples of invalid values are 131 (because 1 occurs twice), 888 (because 8 occurs thrice), -156 (because the number is negative).

The Codebreaker tries to guess N by coming up with a positive integer X which is also made of 3 unique digits.

(Note: Consider the number 843. The digits are 8, 4 and 3. Digit 8 is in the hundreds position, 4 is in the tens position, and 3 is in the ones position).

Your task is to implement a function that will compare EACH digit of X with N, and based on the comparison it should output a corresponding character to denote the comparison result. There are three possible scenarios:

1. If a digit of X matches a digit in N in the same position, the program should output an asterisk '*' character. For example, if N = 843, and X = 825, digit 8 is an exact match – same value, and same position, i.e., hundreds position.
2. If there is a partial match, i.e., a digit of X matches a digit in N but NOT in the same position, the program outputs a plus '+' character. For example, if N = 843, and X = 582, digit 8 has a match but in a different position (hundreds for N, while it is in the tens for X).
3. If there is no match, i.e., a digit of X does not match any of the digits in N, the program outputs a minus '-' character. For example, if N = 843, and X = 267, all digits of X are different from the digits of N.

The following shows several Test Cases and the expected output of the function. The first line shows the value of N, the second line shows the value of X, and the third line is the program's output.

Test Case #1: Test for an exact match for all digits; output should show ***.

```
843
843
***
```

Test Case #2: Test for no match for all digits; output should show three ---.

```
843
267
---
```

Test Case #3: Test for partial match for all digits; output should show three +++.

```
843
384
+++
```

Test Case #4: Test for a combination of three scenarios; output should show *-+.

```
843
824
*-+
```

Test Case #5: Test for another combination of three scenarios; output should show +*-.

```
843
342
+*-
```

Test Case #6: Test for one partial match, and two digits with no match; output should show -+-.

```
843
531
-+-
```

Test Case #7: Test for two exact matches, and a digit with no match; output should be -**.

```
843
943
-**
```

Can you think of other Test Cases? Is it possible to have a *+* as a result? If yes, provide example values of N and X that will produce such a result. If not, explain why such a result cannot be achieved.

A partial program is shown on the next page. Your task is to complete the program, i.e., implement the logic of the function **MasterMind().**

```
#include <stdio.h>

/*
*  Compare X (the guess) with N (the secret code).
*  Pre-condition: Assume that both N and X contain valid 3-digit positive
*     integer value.
*/
void
MasterMind(int N, int X)
{
      /* Implement the body of this function.
          You may declare any number of local variables that you need.
          You may define other functions that can be called inside this function.
      */
}

int
main()
{
      /* You are NOT allowed to change/edit the main() function! */
      int X;
      int Y;

      scanf("%d", &N);
      scanf("%d", &X);

      MasterMind(N, X);
      return 0;
}
```

--- end of this document ---