Programming Exercise 9.3

NSA Decoder, v.0.0

Purpose. Learn that characters are represented in computer memory as whole numbers -- that these are *code numbers* that represent *characters*. Since they are numbers, it should be possible to perform simple math operations with them, as you will experience here.

Pretend that you have been hired by the National Security Agency (NSA) to write a program that decodes scrambled text in a secret text file, and outputs the unscrambled version to the console screen. Someone else will be hired to write a program to scramble the file. In this way, the NSA can put secret messages into text files, scramble the files with your program, and send them as attachments to email. The recipients of the email messages will use the decoder program to read the file and display the original message.

But in advance of writing something to process text files, the first step is to perform a “proof of concept”. This “zeroth” version of the program reads a line of encoded text from the console keyboard, decodes it, and outputs the unscrambled version.

Requirements. Write **`nsaDecoder0.cpp`**. Prompt the user to enter a line of *encoded* text, of any length, with spaces and punctuation. Then "decode" the entered text so that each character is converted to the previous character in the ASCII code. That is, covert 'B' to 'A', '5' to '4', '+' to '*', etc. So the scrambled word " **`Ifmmp`**" will appear as "Hello" when decoded, thus protecting government secrets!

Here is a code sample that you found on the Internet after doing some research. It may be of help in this project:

```
// decode string s by subtracting 1 from the ASCII code of each char
string s = "Ifmmp-!Xpsme"; // a scrambled string
for (int i = 0; i < s.length(); i++) // for each char in the string...
  s[i]--; // reduce its ASCII code by 1
```

This decodes to “Hello, World”.

I looked it up, and **`s[i]`** retrieves the `char` value that's at position **`i`** in the string **`s`**.

Program I/O. Input: a line of encoded text via the console keyboard. Output: The decoded version of the input.

Example. Here's what the output should look like, with user input in blue:

```
Enter a line of text:
J!sfbmmz!mjlf!dpnqvufs!qsphsbnnjoh"
I really like computer programming!
```

HINT: To avoid having to type encoded text, you can copy and paste it at the console input prompt. On a Windows 8.1 PC or Mac, it’s right-click (or control-click), Paste.