

Programming Exercise 9.2

NSA Encoder, 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 scrambles the text in a text file, and rewrites it as a new file. Someone else will be hired to write a program to unscramble the file and print it to the screen. 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 text from the console keyboard, scrambles it, and outputs the scrambled version.

Requirements. Write `NsaEncoder0.java`. Prompt the user to enter a line of text, of any length, with spaces and punctuation. Then “encode” the entered text so that each character is converted to the next character in the ASCII code. That is, convert 'A' to 'B', '4' to '5', '*' to '+', etc. So the word “Hello” will appear as “Ifmmp” in encoded form, 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:

```
// to build encoded string, add 1 to the ASCII code of each char
String sOld = "Hello, World"; // an example
String sNew = ""; // the scrambled version of sOld
for (int i = 0; i < sOld.length(); i++) // for each char in string...
    sNew += (char) (sOld.charAt(i) + 1); // bump by one
```

I looked it up, and `sOld.charAt(i)` retrieves the `char` value that's at position `i` in the string `sOld`.

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

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

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