Caesar used to “encrypt” (i.e., conceal in a reversible way) confidential messages by shifting each letter therein by some number of places based on a key.

Design and implement a program, Caesar, that encrypts messages using Caesar’s cipher.

* Your program must accept a single command-line argument, a non-negative integer. Let’s call it k for the sake of discussion.
* If your program is executed without any command-line arguments or with more than one command-line argument, your program should print an error message of your choice (with printf) and return from main a value of 1 (which tends to signify an error) immediately.
* If any of the characters of the command-line argument is not a decimal digit, your program should print the message Usage: ./caesar key and return from main a value of 1.
* Do not assume that k will be less than or equal to 26. Your program should work for all non-negative integral values of k less than 2^31 - 26. In other words, you don’t need to worry if your program eventually breaks if the user chooses a value for k that’s too big or almost too big to fit in an int. (Recall that an int can overflow.) But, even if k is greater than 26, alphabetical characters in your program’s input should remain alphabetical characters in your program’s output. For instance, if k is 27, A should not become [ even though [ is 27 positions away from A in ASCII, per http://www.asciichart.com/[asciichart.com]; A should become B, since B is 27 positions away from A, provided you wrap around from Z to A.
* Your program must output plaintext: (without a newline) and then prompt the user for a string of plaintext (using get\_string).
* Your program must output ciphertext: (without a newline) followed by the plaintext’s corresponding ciphertext, with each alphabetical character in the plaintext “rotated” by k positions; non-alphabetical characters should be outputted unchanged.
* Your program must preserve case: capitalized letters, though rotated, must remain capitalized letters; lowercase letters, though rotated, must remain lowercase letters.
* After outputting ciphertext, you should print a newline. Your program should then exit by returning 0 from main.