In order to hide messages from their enemy, the Roman Empire used what’s called a shift cipher to scramble the contents of their messages. This works by replacing each letter in a word by a shifted letter, where the ‘key’ for decrypting the message is the shift amount.

For example if the shift key is 2

A becomes C

B becomes D

C becomes E

…

and letters wrap around, so

Y becomes A

Z becomes B

And so on..

So using this shift amount, the phrase **LabVIEW Rocks** becomes **NcdXKGY Tqemu**

Your challenge is to create a LabVIEW program that takes a string input and a shift amount, and outputs the encrypted version. This should be done with the following guidelines:

        - Only letters of the alphabet should be shifted, numbers and punctuation should not change

         -Capital letters must stay capital, and lower case letters must remain lower case

         -The shift amount should be between 1-25

         -You should not have to use any Local/Global variables

**Hint:**

The following LabVIEW block might prove to be very helpful for this challenge:

**Numeric -> Conversion -> String to Byte Array**

**Bonus:**

After experimentation, you have found that the above cipher is fairly weak and simple to crack. To make it stronger you are to implement an algorithm which shifts each letter a different amount. The encryption key will be a passphrase, where each letter in the passphrase represents a number based on its location in the alphabet. For full credit on the bonus this program should be able to both **encrypt** and **decrypt** based on this method, and using the same guidelines as above.

Example:

The passphrase **DOG**represents shift amounts  **4  15  7**  (D=4, O=15, G=7)

If I want to encrypt the phrase **LabVIEW Rocks** with the key **DOG**

**L**gets shifted by 4, and becomes **P**

**a**gets shifted by 15, and becomes **p**

**b**gets shifted by 7, and becomes **i**

once you run out of characters you repeat, so

**V**gets shifted by 4, and becomes **Z**

**I**gets shifted by 15, and becomes **X**

And so on until you get:

**PpiZXLA Ysrrw**

**----**

New Topics:

Arrays

For loops & auto indexing

How a character and a string are represented (ASCII)

Case Structures