**Elad Or – I.D 318689627  
  
Encryption function** - takes a text (string) and a key, turns the text in a **list**, creates a new empty list and an int counter, loops **for each letter in the string (turned list)**, raise the counter by 1 and add that letter to the new list, with the addition of the key multiplied by the counter + key if the **modulo** of the key by the current counter returns 0.

Whitespace and new lines are replaced with gibberish string.

Then we return the new list after we turn it into a string and reverse it to make it more ambiguous.

The **decryption function** basically reverses that same process and will only fully decrypt the input when given the right key.

**Forced decryption** - takes a text **hint** (so it knows what to look for in general) and then a **key search range**.  
A 'for' loop starts and runs the decryption algorithm for \*search range\* times, if a result that contains the hint text exists - it returns the result and the key that worked and then the loop ends, a correct but weak hint can give less accurate results.  
New lines will be replaced with a long random text (ultra slim chance that this exact gibberish text will exist in the text file the user wants to decrypt so there shouldn't be a problem with that.  
If it doesn't find a string that contains the hint text, it returns 'no plausible result found’.  
  
If any input is missing – it will let the user know in the output box.

If the input was given correctly – every encryption/decryption combo (for example – import from file and force decrypt to file) should work with no traceback errors or bugs.