CS2261 Fall 2018 Hauschild

PROGRAMMING ASSIGNMENT 5:

[100 pts] You have become paranoid that the lizardman have taken over the world government and are spying on you. Since they obviously took over all corporations first, you cannot trust commercial encryption and so have decided to write your own encryption software to communicate with the resistance.

In this assignment, you will be writing two programs. One to encrypt a file and another to read an encrypted file.

For the first program, prompt the user for a byte in the range of -128 to 127. This is your key for the code. Then prompt them for a filename (for this example, suppose they entered "plaintext").

Your task is then to encrypt this file using your key and save it to another file with the same name but an additional file extension, ".encrypted". So, read this filename byte by byte. As you read it by byte, add the key value to it and save it by byte another file, named plaintext.encrypted.

For the second program, prompt the user for a byte in the range of -128 to 127 and a file to decrypt. Use this key to decrypt a file and save the now unencrypted file to the file given, with the additional extension .decrypted.

So for example, suppose the user runs the first program on a file called "secretrevolutionplans" with the key of 5, the output would be a file "secretrevolutionplans.encrypted" with all the bytes having 5 added to them. After decrypting it with the decrypt file, you would have a file "secretrevolutionplans.encrypted.decrypted" that should have identical contents to 'secretrevolutionplans".

Note 1: You MUST use byte reads, do not simply read characters and add to them for this.
Note 2: Bytes are stored twos complement in java, so their range is -128 to 127. If you try and add two bytes directly, like this:
byte b=3;c=2;
byte a = b + c;
java will complain, as the addition operator returns an integer. To fix this, you must add them as follows:
byte a = (byte)(b+c);
Note 3: If actually planning revolution, might not want to name your file "secretrevolutionplans".

Encryption scheme comments: Obviously this encryption scheme is very poor. Thankfully the lizardmen are probably pretty stupid. However, if you wish to increase the power of your encryption, feel free to use your own encryption scheme. If you do use your own scheme, please document it in your code and it should at least be as strong as the default scheme.

[20 extra credit] For 20 extra credit, write a 3rd program that reads in a text and attempts to break the code used in the file (assuming the encryption scheme is the simple additive version that we gave above). To do this automatically, it will have to keep trying various keys and then make some determination of whether or not the new file is "unencrypted". This is probably only possible if the original file was in text. I leave this up to you as to how you might try to do this. This program should then output what key was used and then output a file with the additional file extension ".broken" with the decrypted file. Please comment your attempts in your documentation if you do this.