

Encode/Decode OTP with LCG: Programming Project CMSC 456 Sample Inputs/Outputs and Testing Instructions

1. Sample unencrypted text:

```
helloworld
```

2. Corresponding encrypted text(with $x_0 = 1, A = 1000, B = 2000, M = 9001$):

```
38 05 58 79 79 05 47 98 68 30
```

Save the examples in text files; in the sample commands below, they are saved in files named `encrypted.txt` and `unencrypted.txt`. To test your code, move the sample text files to the same directory as your code, navigate to that directory in your terminal, and use the following commands (which should work in either a unix shell or in Windows command prompt, NOT powershell which may be your vscode default):

1. To test encode:

```
python3 encode.py 1 1000 2000 9001 < unencrypted.txt > encrypted.txt
```

and see that your program's output `encrypted.txt` exactly matches the given one (case-insensitive).

2. To test decode:

```
python3 decode.py 1 1000 2000 9001 < encrypted.txt > unencrypted.txt
```

and see that your program's output `unencrypted.txt` exactly matches the given one (case-insensitive).

You may need to modify the filename in these commands (`encode.py` or `decode.py`) to whatever you have named yours. These examples are in Python, so if you're testing another language, change "python3" to the appropriate command ("javac" for Java, "ruby" for Ruby, "a.out" for C and C++, etc).

This sample is NOT exhaustive. I would recommend modifying the input to encode (modify `unencrypted.txt`) by adding whatever your program may face, which could be newlines, special characters, numbers, spaces, tabs, etc. Your output should not include these characters you add to the input, unless you add more letters. For decode, you can assume your input will look exactly like the one above— no extra spaces, lines, characters, etc.