Homework1

Deadline: 27 May(2359)

Objectives

- To understand and modify a simple Python 3 script to read, and parse arguments
- To write a shift cipher for ASCII and binary

Shift Cipher

Python Tips

Please take care to validate the arguments provided carefully, which is good practice to prevent bugs and security vulnerabilities. In particular, try to ensure the following in your submissions:

- For each argument, check if it has the correct type (e.g. string or integer)
- If required, ensure that the input comes from a specific range or set of inputs, e.g. is a number >= 0 and < 256
- It is good practice to provide a default value for parameters, in case they are not provided by the user

You will need to use the Python `argparse` module. Some references:

- Longer tutorial
- Documentation

Part I: Shift Cipher for printable input

Write a Python script that can be run from the command prompt or shell as the following:

```
python3 ex1.py -i [input filename] -o [output filename] -k [key] -m [mode]
```

The Python script should be able to

1. Load the the text file given `sherlock.txt`
2. Encrypt it with a given key
3. Output the cipher text file
4. Decrypt your own cipher text file

In your submission, you will need to check whether the text remains the same after the process of encryption and decryption.

For this part of the exercise we will use the string module and in particular we will focus on the `string.printable` string as our valid alphabet.
About key `-k`:

- The key must be between `1` and `len(string.printable)-1` (inclusive), otherwise an error message is produced.
- The key specifies how many characters the original letter should be shifted.
- The output and input space for characters should include the set specified by string.printable in Python.
- For encryption, the key is added to the original letter.
- For decryption, the key is subtracted from the original letter.

```
# example encryption
E("A", k=10) = 65 + 10 = 75 = "K"
# example decryption
D("K", k=10) = 65 = "A"
```

About mode `-m`:

- The mode must either be `d` (decryption), `e` (encryption).
- Input should be **case insentitive**
- When there is an invalid input, an appropriate error message should be produced to indicate the error and show the available options

A starter template file for `ex1.py` has been provided - feel free to use it or start from scratch.

Part II: Shift Cipher for binary input

Based on `ex1.py`, write a Python script that can be run from the command prompt or shell as the following:

```
python3 ex2.py -i [input filename] -o [output filename] -k [key] -m [mode]
```

The Python script should be able to

1. Load the the text file given `sherlock.txt`
2. Encrypt or decrypt it with a given key

In your submission, you will need to check whether the text remains the same after the process of encryption and decryption.

- The key must be between 0 and 255 (8-bit / 1-byte integer), otherwise an appropriate error message is produced.
- The input should be interpreted as sequence of bytes.
- The key specifies how many characters each original byte should be shifted.
- Consider the full extended-ASCII range of 256 values in your encryption and decryption implementation
- For encryption, the key is added to the original byte.
- For decryption, the key is subtracted from the original byte.
- Hint: `bytearray` from Python 3 can be helpful.
- For example we can see the result of a binary shift using hexstring representation:

```
0x0b = 10 = 0b001010

p = 0x01, k = 0x0b
E(p, k) = c = 0x01 + 0x0b = 0x0c
D(c, k) = 0x01
```

Part III: Break Shift Cipher of flag

Use your shift cipher script from the Part II to find the plaintext (decrypted file) corresponding to the provided flag file.

Hints:

- You can use `file` command line tool to determine the type of a binary file.
- The file is not a `txt` file.
- There are only a finite number of keys...

Submission

eDimension Submission

Submission ground rules:

- Please make sure to indicate your name and student ID in each of the graded submission files

Lab 1 submission:

Upload a **zip file** with the following:

- `ex1.py`
- `ex2.py`
- Decrypted `flag` file (with the correct file extension - e.g. `.txt`)
- Optional: `report` Jupyter Notebook (with the outputs saved)