Installation

- We implement algorithms by `python` so make sure that you install `python` before executing the script
- Note that we use libraries `hashlib`, if available try to install:

```
pip install hashlib
```

Getting started

```
python algorithm.py
```

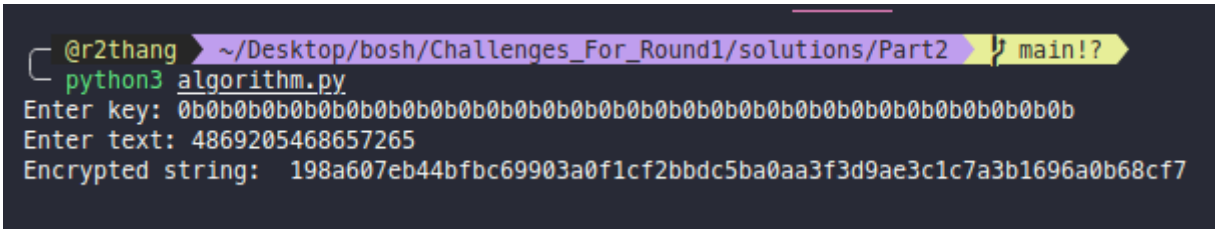
- Enter your key, text and got the output of the algorithm

Example

- For example, use the `Test vector`

```
Key: 0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b (32
bytes)
Data: 4869205468657265
```

- Output from the algorithm:
  `198a607eb44bfbc69903a0f1cf2bbdc5ba0aa3f3d9ae3c1c7a3b1696a0b68cf7`
- Our output:



> Same as above

Explanation

- We following the description of the problem:

- Step 1: If the length of K = B: set Ko = K. Go to step 4.
- Step 2: If the length of K > B: hash K to obtain an L byte string, then append (B-L) zeros to create a B-byte string Ko (i.e., Ko = H(K) Il 00 .00). Go to step 4.
- Step 3: If the length of K < B: append zeros to the end of K to create a B-byte string Ko (e.g., if K is 20 bytes in length and B = 64, then K will be appended with 44 zero bytes x'00').
- Step 4: Exclusive-Or Ko with ipad to produce a B-byte string: K0 (+) ipad.
- Step 5: Append the stream of data 'text to the string resulting from step 4: (Ko (+) ipad) || text.
- Step 6: Apply H to the stream generated in step 5: H((K (+) ipad) || text).
- Step 7: Exclusive-Or K0 with opad: K0 (+) opad.
- Step 8: Append the result from step 6 to step 7: (Ko (+) opad) || H((Ko (+) ipad) || text).
- Step 9: Apply H to the result from step 8: H (K6 (+) opad OIL H((K (+) ipad) || text)).

> All function are already noted from the file algorithm.py