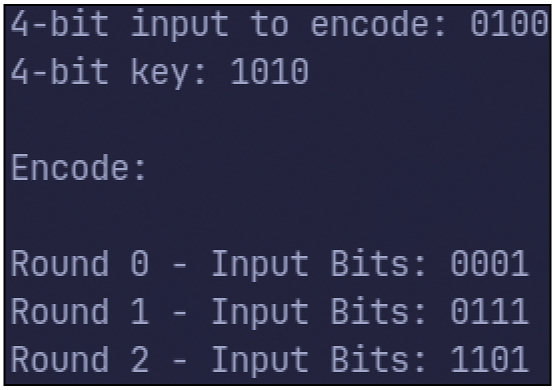
Lab 07 - Block Ciphers Baby Fiestel (DES)

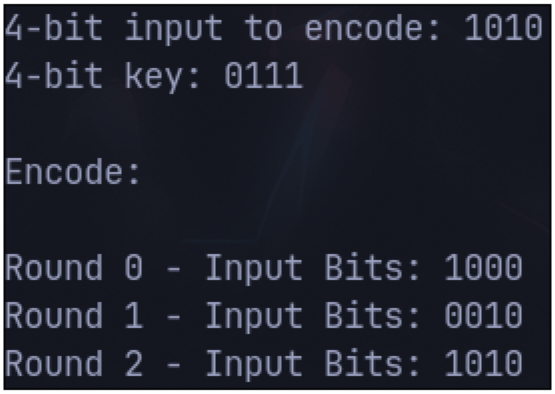
Part 01

1. **Upload python code for babyFeistel cipher**
   (10 points)

2. **Final 4-bit ciphertext**
   (10 points)

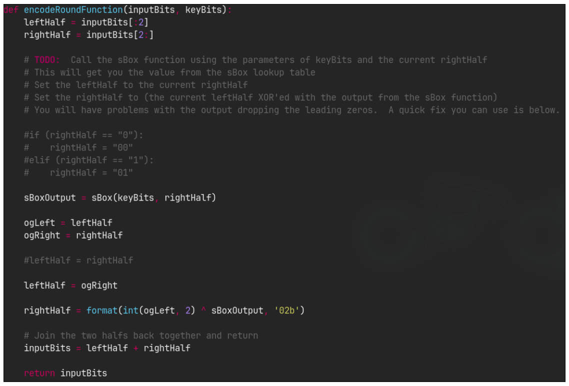
   - 

     ```
     4-bit input to encode: 0100
     4-bit key: 1010

     Encode:

     Round 0 - Input Bits: 0001
     Round 1 - Input Bits: 0111
     Round 2 - Input Bits: 1101
     ```

   - 

     ```
     4-bit input to encode: 1010
     4-bit key: 0111

     Encode:

     Round 0 - Input Bits: 1000
     Round 1 - Input Bits: 0010
     Round 2 - Input Bits: 1010
     ```

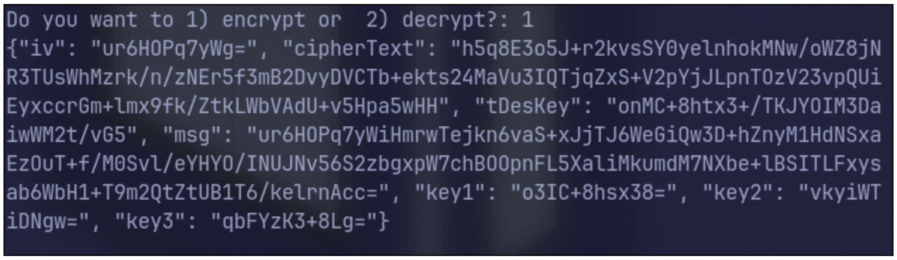
3. **Screenshot of encrypting**
   (10 points)

- 

```
def encodeRoundFunction(inputBits, keyBits):
    leftHalf = inputBits[:2]
    rightHalf = inputBits[2:]

    # TODO:  Call the sBox function using the parameters of keyBits and the current rightHalf
    # This will get you the value from the sBox lookup table
    # Set the leftHalf to the current rightHalf
    # Set the rightHalf to (the current leftHalf XOR'ed with the output from the sBox function)
    # You will have problems with the output dropping the leading zeros.  A quick fix you can use is below.

    #if (rightHalf == "0"):
    #    rightHalf = "00"
    #elif (rightHalf == "1"):
    #    rightHalf = "01"

    sBoxOutput = sBox(keyBits, rightHalf)

    ogLeft = leftHalf
    ogRight = rightHalf

    #leftHalf = rightHalf

    leftHalf = ogRight

    rightHalf = format(int(ogLeft, 2) ^ sBoxOutput, '02b')

    # Join the two halfs back together and return
    inputBits = leftHalf + rightHalf

    return inputBits
```

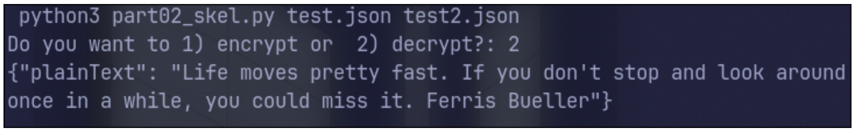
Part 02

4. **Include a screenshot of the json dumps result when encrypting the part02_plainText.json in your lab report. Must be 7 fields**
   (10 points total)

   - 

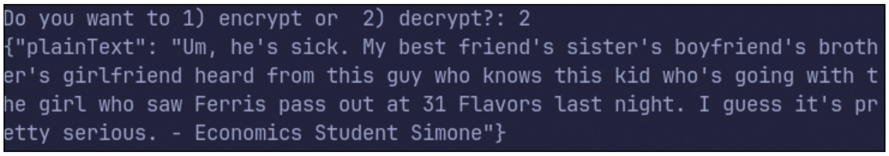
```
Do you want to 1) encrypt or  2) decrypt?: 1
{"iv": "ur6HOPq7yWg=", "cipherText": "h5q8E3o5J+r2kvsSY0yelnhokMNw/oWZ8jN
R3TUsWhMzrk/n/zNEr5f3mB2DvyDVCTb+ekts24MaVu3IQTjqZxS+V2pYjJLpnTOzV23vpQUi
EyxccrGm+lmx9fk/ZtkLWbVAdU+v5Hpa5wHH", "tDesKey": "onMC+8htx3+/TKJYOIM3Da
iwWM2t/vG5", "msg": "ur6HOPq7yWiHmrwTejkn6vaS+xJjTJ6WeGiQw3D+hZnyM1HdNSxa
EzOuT+f/M0Svl/eYHYO/INUJNv56S2zbgxpW7chBOOpnFL5XaliMkumdM7NXbe+lBSITLFxys
ab6WbH1+T9m2QtZtUB1T6/kelrnAcc=", "key1": "o3IC+8hsx38=", "key2": "vkyiWT
iDNgw=", "key3": "qbFYzK3+8Lg="}
```

5. **Include a screenshot of the json dumps result when decrypting the encrypted part02_plainText.json you just encrypted in your lab report.**
   (10 points total)

   - 

```
 python3 part02_skel.py test.json test2.json
Do you want to 1) encrypt or  2) decrypt?: 2
{"plainText": "Life moves pretty fast. If you don't stop and look around
once in a while, you could miss it. Ferris Bueller"}
```

6. **Include a screenshot of the json dumps result when decrypting part02_cipherText.json in your lab report.**
   (10 points total)

   - 

```
Do you want to 1) encrypt or  2) decrypt?: 2
{"plainText": "Um, he's sick. My best friend's sister's boyfriend's broth
er's girlfriend heard from this guy who knows this kid who's going with t
he girl who saw Ferris pass out at 31 Flavors last night. I guess it's pr
etty serious. - Economics Student Simone"}
```

7. Upload python code for part 02.
   (10 points)

8. What block cipher mode are you using in this implementation of Triple DES?
   (10 points)

- It tells you within this line
  cipher = DES3.new(tDesKey, DES3.MODE_CFB)
  Which is Cipher Feedback Mode.

9. Explain why, even though you create 24 bytes of key, the effective key size is 112 bits. (10 points)
   - The keying process is done by using 56-bit keys. The three keys 1,2,3 are different by the middle key being the same as the first key and the third key being the same as the middle key. This reduces the choices because there are only 2^56 possible keys. Since the middle stage acts as an decryption, it reduces the key to 112 because the first and third stages must use the same key.

10. Historically, there has been a way to use a single key in Triple DES where you set key1=key2=key3. Why would you want to do that? What is the effective key size when key1=key2=key3?
    (10 points)
    - When key1=key2=key3 it reduces 3DES to DES. This is because the middle stage is decryption, and pretty much undoes the encryption performed by the first step and doesn't provide additional security and the effective key size is still only 56 bits. So you might as well just use DES than go through a slower process with the same result.