Lab 03 (4 hrs): Symmetric Encryption

Part 1 (1 hrs):

Program 1: 3DES

In this program, you are required to implement the 3DES algorithm using the provided encrypt and decrypt function of DES. The encrypt and decrypt method of 3DES should also be pure functions, i.e. without side effects.

Your program does the following:

- Read a hex string from the console input. The string represents the plaintext bytes as a hex string.
- Read a hex string from the console input. The string represents the first key bytes as a hex string.
- Read a hex string from the console input. The string represents the second key bytes as a hex string.
- Read a hex string from the console input. The string represents the third key bytes as a hex string.
- Encrypt the plaintext with the three keys.
- Print the ciphertext bytes as a hex string.
- Decrypt the ciphertext with the three keys.
- Print the plaintext bytes after decryption as a hex string.

Example Input & Output

Input:

```
8787878787878787
133457799bbcdff1
0e329232ea6d0d73
133457799bbcdff1
```

Output:

```
e98a0b8e59b3eeb7
8787878787878787
```

Part 2 (3 hrs):

Program 2: AES

Modes of operations allow you to encrypt more data than the block size of your symmetric block cipher. Example: `CBC`.

In this program, you are required to demonstrate the `AES-256-CBC` algorithm **with a third-party crypto library**, `pycryptodome`. Recall that you must provide a corresponding `requirements.txt` file if any third party libraries are involved in the code.

Your program does the following:

- Read a text string from the console input.
- Encode the text string with `utf-8` encoding, as the plaintext bytes.

- Pad the plaintext bytes with `pkcs7` algorithm.
- Print the padded bytes as a hex string.
- Read a Base64 string from the console input. The string represents the key bytes as a hex string. If the length of key bytes is not expected, abort the program with a Python code `raise Exception('key length mismatch')`
- Read a Base64 string from the console input. The string represents the IV bytes as a hex string. If the length of IV bytes is not expected, abort the program with a Python code `raise Exception('IV length mismatch')`
- Encrypt the padded plaintext bytes with the key and IV.
- Print the ciphertext bytes as a Base64 string.
- Decrypt the ciphertext bytes with the key and IV.
- If the decrypted plaintext bytes are identical with the unpadded plaintext bytes, print `identical`. Otherwise, print `not identical`.
- Unpad the decrypted plaintext bytes with `pkcs7` algorithm.
- Print the unpadded bytes as a hex string.
- Decode the unpadded bytes with `utf-8` encoding, and print the decoded text string.

Example Input & Output

Input:

```
I don't like deadbeef. 你呢？
1UO7ZnmwcT7KtScS2hAZV+aZ1Gk95HPK1EqcXT6rqoU=
6GXIzJ0GD/76WkTtgmaDYQ==
```

Output:

```
4920646f6e2774206c696b652064656164626565662e20e4bda0e591a2efbc9f10101010101010101010101010
1010101010
c0LWy2BUg949eMO+G8NgxUzKVNNFys8EzavYFhP0Tc/mZM/UVVe4E3b34cEyu1Ze
not identical
4920646f6e2774206c696b652064656164626565662e20e4bda0e591a2efbc9f
I don't like deadbeef. 你呢？
```

Note: the first line of the example input is consisting of the following 26 characters:

I		d	o	n	'	t		l	i
k	e		d	e	a	d	b	e	e
f	.		你	呢	?				