Advanced Security
Lab 7
Student Name: Jonathan Riordan
Student ID: C13432152

Part 1.

Code:

```
'''
Advanced Security
Student Name: Jonathan Riordan
Student ID: C13432152

Lab 7 - Part 1
'''
Advanced Security
Student Name: Jonathan Riordan
Student ID: C13432152

Lab 7 - Part 1
'''
import base64
from Crypto.Cipher import DES

def addPadding(newText):
        length =  8 - (len(newText) % 8)
        newText += "\x00"*(length)
        return newText

def chunks(longdata, n):
        for i in range(8, len(longdata),n):
                yield longdata[i:i +n]

iv = "00000000"
plain_text = "AAAABBBBCCCCD"
plain_text_padding = addPadding(plain_text)
datasource = dict(enumerate(list(chunks(plain_text_padding, 8)), start = 0))

print str(datasource)

hash = iv

for d in datasource:
        des = DES.new(datasource[d], DES.MODE_ECB)
        cipher_text = des.encrypt(hash)
        hash = "".join(chr(ord(x) ^ ord(y)) for x ,y in zip(hash, cipher_text))

print "Plaintext: " + plain_text
print "hash base 16 encoded: " + str(map(''.join, zip(*[iter(base64.b16encode(hash))]*16)))
```

Output:

```
{0: 'CCCCD\x00\x00\x00'}
Plaintext: AAAABBBBCCCCD
hash base 16 encoded: [‘2FA197D2A2D3F976']
```

Part 2.

Code:

```
'''
Advanced Security
Student Name: Jonathan Riordan
Student ID: C13432152
```

```
Lab 7 - Part 2
'''
import hashlib
import hmac
from hashlib import md5

key = "FACEBOOK"
plaintext = "AAAABBBBCCCC"
hash = hmac.new(key, plaintext, md5).hexdigest()
# Compare the output of the two hashes.
print hash
print hmac.compare_digest(hmac.new(key, plaintext, md5).hexdigest(), hash)
```

Output:

```
bdb45f26133aabe937bc0a97c6317054
True
```