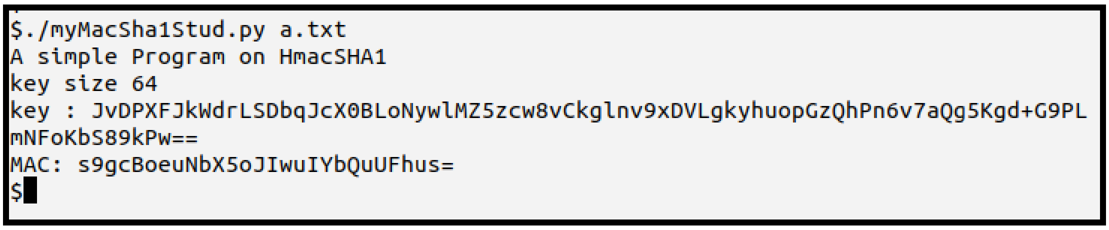
Objectives: By the end of this practical exercise, the students should be able to:

- Apply hmac object from the hashlib package for keyed-Hashing for Message operations.
- Understand the application aspect of Message Authentication Code functions

When completed, the python program will produce the following output.



```
$./myMacSha1Stud.py a.txt
A simple Program on HmacSHA1
key size 64
key : JvDPXFJkWdrLSDbqJcX0BLoNywlMZ5zcw8vCkglnv9xDVLgkyhuopGzQhPn6v7aQg5Kgd+G9PL
mNFoKbS89kPw==
MAC: s9gcBoeuNbX5oJIwuIYbQuUFhus=
$
```

Note: The program will produce different message authentication code depending on the supplied key and the content of the file.

Instructions:

1. Download the python program from the Black Board:

   myMacSha1Stud_skel.py

Task Requirement:

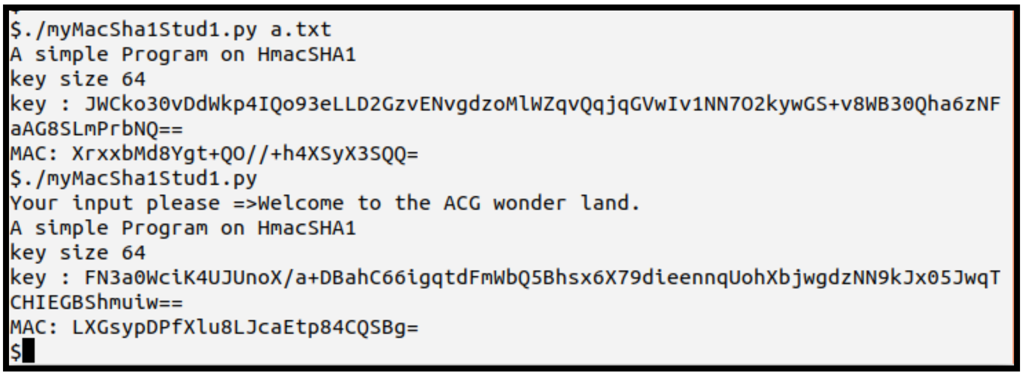
Study the python code and comments in the “myMacSha1Stud_skel.py” (incomplete) program. You are required to complete the above python program to produce the message authentication code (MAC) for any input text file.

Hint:

```
import sys
# main program starts here
argc = len(sys.argv)
if argc != 2:
    print("Usage : {0} <file name>".format(sys.argv[0]))
    exit(-1)
try:
    with open(sys.argv[1]) as f:
        content=f.read()    # read in the entire text file
        print("A simple Program on HmacSHA1")
        keysize=hmac.HMAC.blocksize # retrieve the default block size
        print("key size {0}".format(keysize))
        # insert your code here to generate a random key

        # display the key in base64 encoded bytes in UTF8 format
        print("key : {0}".format(base64.b64encode(key).decode()))
        # insert your code here to instantiate a sha1 hmac object, hma .

        # insert your code here to use hma to compute the hmac of content.

        # insert your code here to display the MAC digest in base64
        # encoded bytes in UTF8 format

except:
    print("Invalid file argument!")
```

More Challenging Tasks:

1. Modify the above program so that it will produce the MAC value for any input of phrase or text file.



```
$./myMacSha1Stud1.py a.txt
A simple Program on HmacSHA1
key size 64
key : JWCko30vDdWkp4IQo93eLLD2GzvENvgdzoMlWZqvQqjqGVwIv1NN7O2kywGS+v8WB30Qha6zNF
aAG8SLmPrbNQ==
MAC: XrxxbMd8Ygt+QO//+h4XSyX3SQQ=
$./myMacSha1Stud1.py
Your input please =>Welcome to the ACG wonder land.
A simple Program on HmacSHA1
key size 64
key : FN3a0WciK4UJUnoX/a+DBahC66igqtdFmWbQ5Bhsx6X79dieennqUohXbjwgdzNN9kJx05JwqT
CHIEGBShmuiw==
MAC: LXGsypDPfXlu8LJcaEtp84CQSBg=
$
```

2. Based on the program in question 1 above, modify the program so that it uses HmacSHA256, HmacSHA512 hash functions to produce the respective hash value for any input of phrase or text file.