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

- Apply hash related classes / methods from the standard Python library (hashlib).
- Understand the application aspect of hash functions

EXERCISE 1. "md5sum" is a GNU text utility from Free Software Foundation, Inc. Use this external program to compute the MD5 digest of a file.

EXERCISE 2. Write a Python program `myMd5Stud.py` to produce the MD5 digest for any input file. The MD5 output from your program should match that from md5sum

Below is a sample output of Exercise 1 and 2.

```
$md5sum a.txt
82642eccc6df18237de0f228e52de538  a.txt
$./myMd5Stud.py a.txt
A Simple Program on MD5
MD5 Hex => 82642eccc6df18237de0f228e52de538
End of Program
$
```

Note: The program will produce different message digest when the content of the file has been changed.

Hint:

```
#!/usr/bin/env python3
#ST2504 - ACG Practical - myMd5Stud_skel.py
# Template for myMd5Stud.py
import sys
import hashlib
# 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],"r") as f:
        content = f.read()
        # instantiate your hash object here

        # update the hash object with the file content (in bytes!)
here
        # Retrieve and print the hex string of the message digest.
        print("A Simple Program on MD5")
        # insert your code here

        print("End of Program")
    f.close()
except:
    print("Invalid file argument!")
```

EXERCISE 3. Modify the program in Exercise 2 to use SHA-256 hash function to produce a 256-bit hash value for any input of phrase or text file.

For example: A sample run of it against sha256sum (the GNU Util)

```
$sha256sum a.txt
ad11106d9ccb6f4566d8a0ff3e4b0a3a280646cbaf1c534316294fa92bc04e3d  a.txt
$./mySha256Stud.py a.txt
A Simple Program on Sha256
MD5 Hex => ad11106d9ccb6f4566d8a0ff3e4b0a3a280646cbaf1c534316294fa92bc04e3d
End of Program
$
```

End of Practical