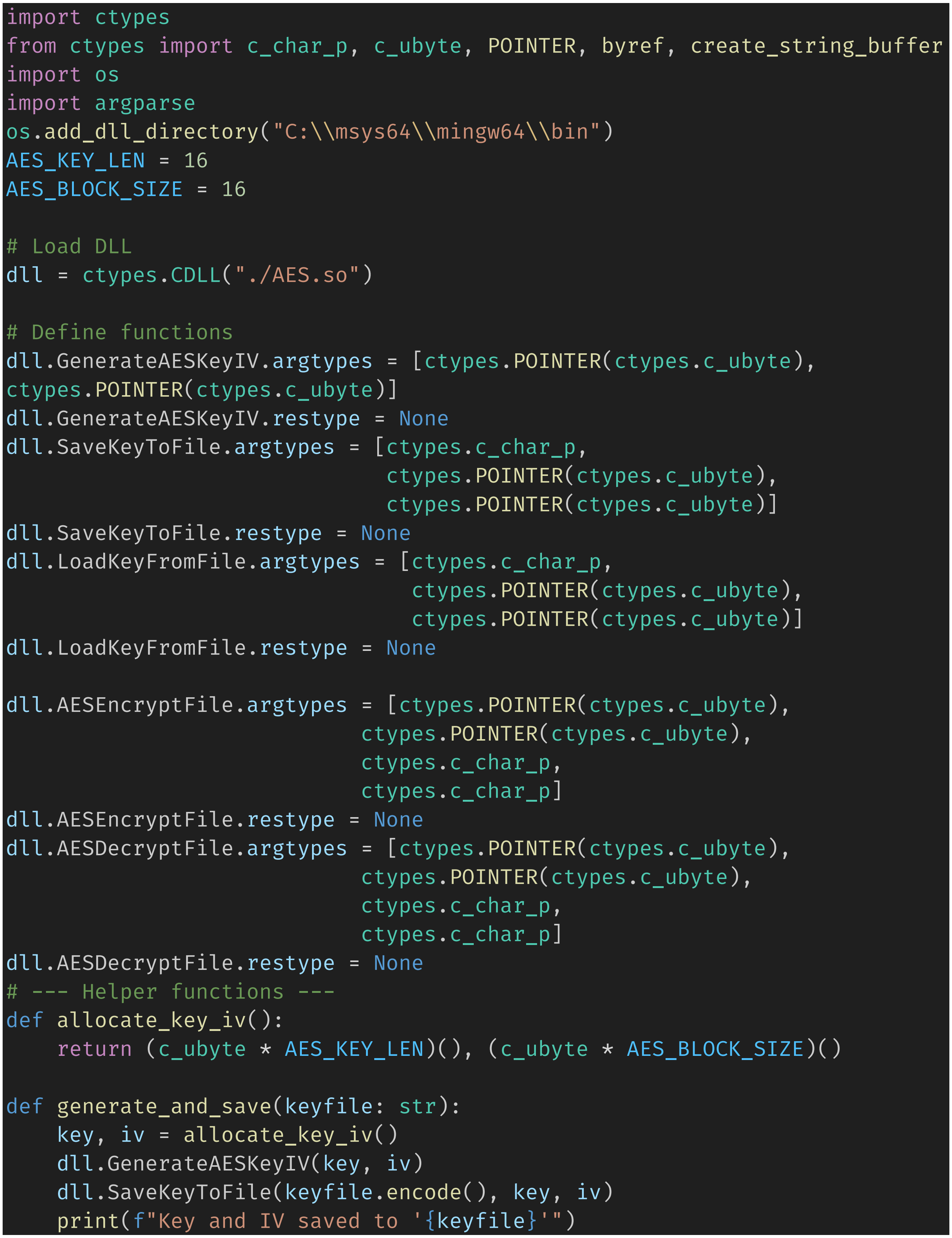
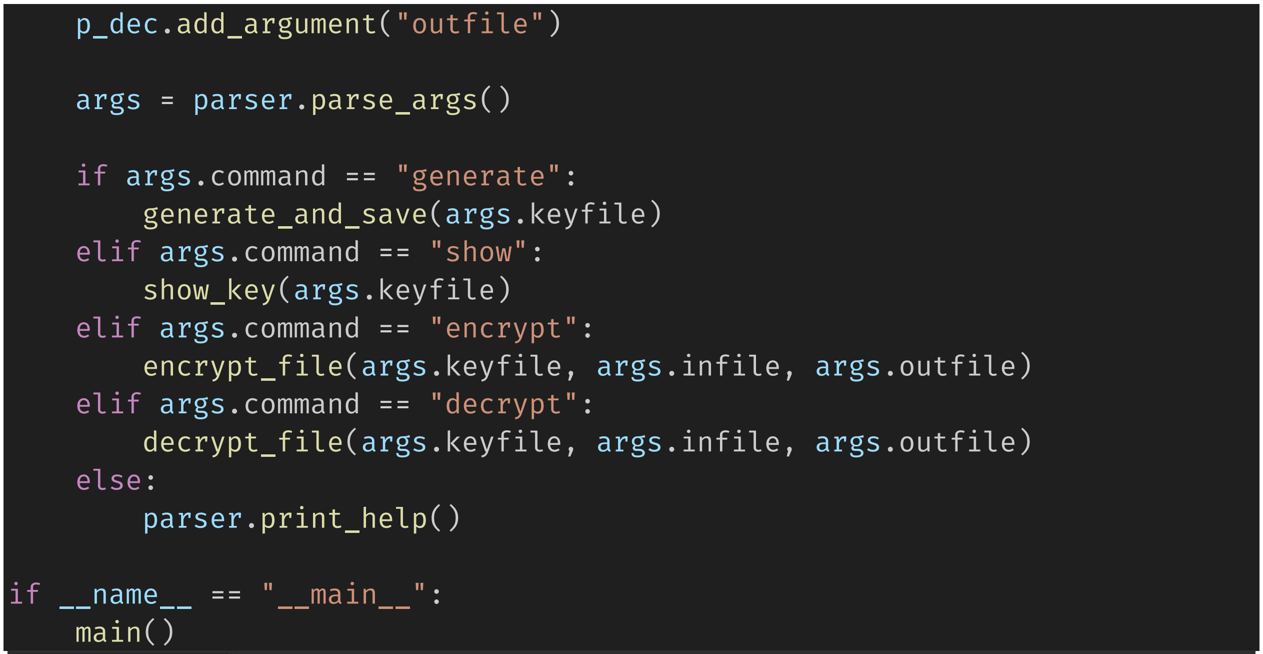
Nguyễn Văn Hưng – 23520569

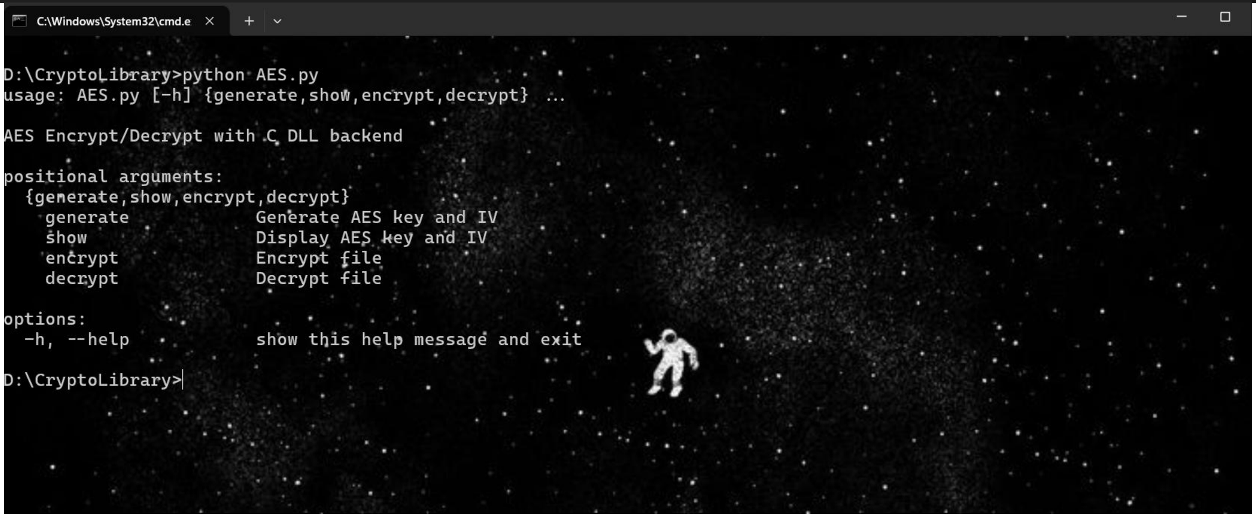


```
import ctypes
from ctypes import c_char_p, c_ubyte, POINTER, byref, create_string_buffer
import os
import argparse
os.add_dll_directory("C:\\msys64\\mingw64\\bin")
AES_KEY_LEN = 16
AES_BLOCK_SIZE = 16

# Load DLL
dll = ctypes.CDLL("./AES.so")

# Define functions
dll.GenerateAESKeyIV.argtypes = [ctypes.POINTER(ctypes.c_ubyte),
ctypes.POINTER(ctypes.c_ubyte)]
dll.GenerateAESKeyIV.restype = None
dll.SaveKeyToFile.argtypes = [ctypes.c_char_p,
                              ctypes.POINTER(ctypes.c_ubyte),
                              ctypes.POINTER(ctypes.c_ubyte)]
dll.SaveKeyToFile.restype = None
dll.LoadKeyFromFile.argtypes = [ctypes.c_char_p,
                                ctypes.POINTER(ctypes.c_ubyte),
                                ctypes.POINTER(ctypes.c_ubyte)]
dll.LoadKeyFromFile.restype = None

dll.AESEncryptFile.argtypes = [ctypes.POINTER(ctypes.c_ubyte),
                            ctypes.POINTER(ctypes.c_ubyte),
                            ctypes.c_char_p,
                            ctypes.c_char_p]
dll.AESEncryptFile.restype = None
dll.AESDecryptFile.argtypes = [ctypes.POINTER(ctypes.c_ubyte),
                            ctypes.POINTER(ctypes.c_ubyte),
                            ctypes.c_char_p,
                            ctypes.c_char_p]
dll.AESDecryptFile.restype = None
# --- Helper functions ---
def allocate_key_iv():
    return (c_ubyte * AES_KEY_LEN)(), (c_ubyte * AES_BLOCK_SIZE)()

def generate_and_save(keyfile: str):
    key, iv = allocate_key_iv()
    dll.GenerateAESKeyIV(key, iv)
    dll.SaveKeyToFile(keyfile.encode(), key, iv)
    print(f"Key and IV saved to '{keyfile}'")
```

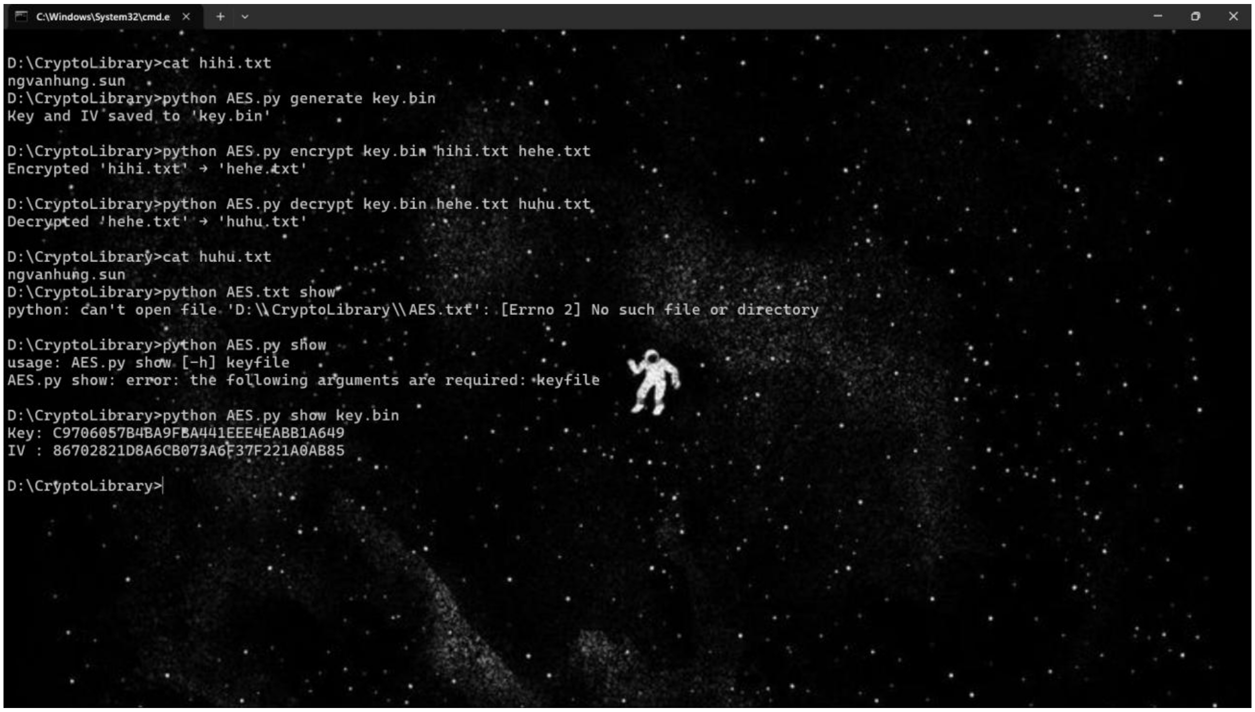
```
def load_key_iv(keyfile: str):
    key, iv = allocate_key_iv()
    dll.LoadKeyFromFile(keyfile.encode(), key, iv)
    return key, iv

def show_key(keyfile: str):
    key, iv = load_key_iv(keyfile)
    print("Key:", bytes(key).hex().upper())
    print("IV :", bytes(iv).hex().upper())

def encrypt_file(keyfile: str, infile: str, outfile: str):
    key, iv = load_key_iv(keyfile)

    dll.AESEncryptFile(key, iv, infile.encode(), outfile.encode())
    print(f"Encrypted '{infile}' → '{outfile}'")

def decrypt_file(keyfile: str, infile: str, outfile: str):
    key, iv = load_key_iv(keyfile)
    dll.AESDecryptFile(key, iv, infile.encode(), outfile.encode())
    print(f"Decrypted '{infile}' → '{outfile}'")
def main():
    parser = argparse.ArgumentParser(description="AES Encrypt/Decrypt with
C DLL backend")
    subparsers = parser.add_subparsers(dest="command")

    # generate
    p_gen = subparsers.add_parser("generate", help="Generate AES key and
IV")
    p_gen.add_argument("keyfile")

    # show
    p_show = subparsers.add_parser("show", help="Display AES key and IV")
    p_show.add_argument("keyfile")

    # encrypt
    p_enc = subparsers.add_parser("encrypt", help="Encrypt file")
    p_enc.add_argument("keyfile")
    p_enc.add_argument("infile")
    p_enc.add_argument("outfile")

    # decrypt
    p_dec = subparsers.add_parser("decrypt", help="Decrypt file")
    p_dec.add_argument("keyfile")
    p_dec.add_argument("infile")
```



```python
    p_dec.add_argument("outfile")

    args = parser.parse_args()

    if args.command == "generate":
        generate_and_save(args.keyfile)
    elif args.command == "show":
        show_key(args.keyfile)
    elif args.command == "encrypt":
        encrypt_file(args.keyfile, args.infile, args.outfile)
    elif args.command == "decrypt":
        decrypt_file(args.keyfile, args.infile, args.outfile)
    else:
        parser.print_help()

if __name__ == "__main__":
    main()
```



```
D:\CryptoLibrary>python AES.py
usage: AES.py [-h] {generate,show,encrypt,decrypt} ...

AES Encrypt/Decrypt with C DLL backend

positional arguments:
  {generate,show,encrypt,decrypt}
    generate            Generate AES key and IV
    show                Display AES key and IV
    encrypt             Encrypt file
    decrypt             Decrypt file

options:
  -h, --help            show this help message and exit

D:\CryptoLibrary>
```



```
D:\CryptoLibrary>cat hihi.txt
ngvanhung.sun
D:\CryptoLibrary>python AES.py generate key.bin
Key and IV saved to 'key.bin'

D:\CryptoLibrary>python AES.py encrypt key.bin hihi.txt hehe.txt
Encrypted 'hihi.txt' → 'hehe.txt'

D:\CryptoLibrary>python AES.py decrypt key.bin hehe.txt huhu.txt
Decrypted 'hehe.txt' → 'huhu.txt'

D:\CryptoLibrary>cat huhu.txt
ngvanhung.sun
D:\CryptoLibrary>python AES.txt show
python: can't open file 'D:\\CryptoLibrary\\AES.txt': [Errno 2] No such file or directory

D:\CryptoLibrary>python AES.py show
usage: AES.py show [-h] keyfile
AES.py show: error: the following arguments are required: keyfile

D:\CryptoLibrary>python AES.py show key.bin
Key: C9706057B4BA9FBA441EEE4EABB1A649
IV : 86702821D8A6CB073A6F37F221A0AB85

D:\CryptoLibrary>
```