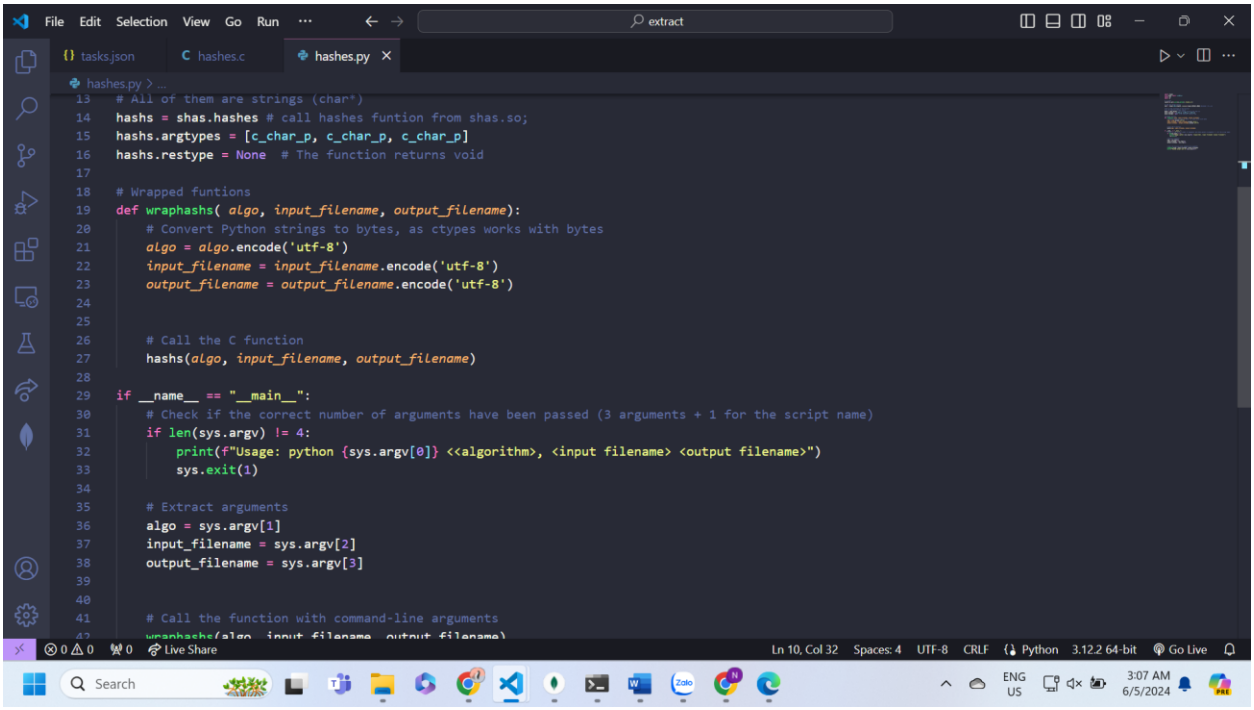
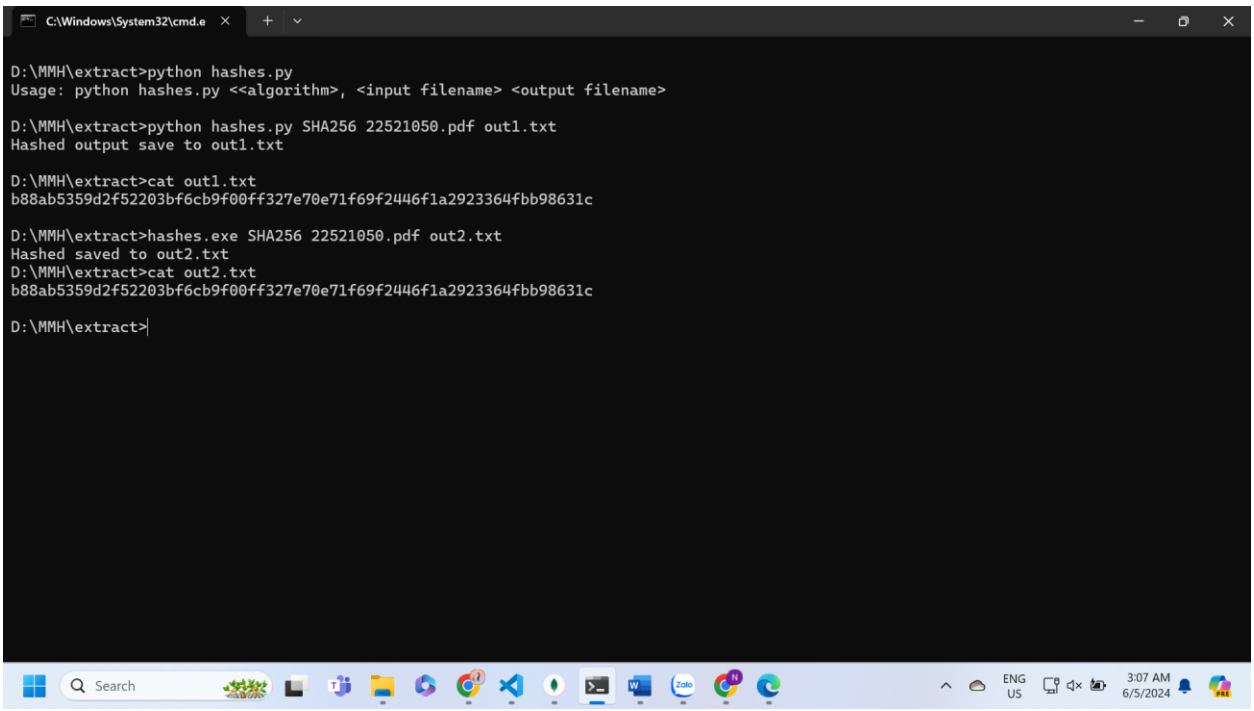

File Edit Selection View Go Run
extract
tasks.json
hashes.c
hashes.py
hashes.py > ...
# All of them are strings (char*)
hashs = shas.hashes # call hashes funtion from shas.so;
hashs.argtypes = [c_char_p, c_char_p, c_char_p]
hashs.restype = None # The function returns void
# Wrapped funtions
def wraphashs(algo, input_filename, output_filename):
# Convert Python strings to bytes, as ctypes works with bytes
algo = algo.encode('utf-8')
input_filename = input_filename.encode('utf-8')
output_filename = output_filename.encode('utf-8')
# Call the C function
hashs(algo, input_filename, output_filename)
if __name__ == "__main__":
# Check if the correct number of arguments have been passed (3 arguments + 1 for the script name)
if len(sys.argv) != 4:
print(f"Usage: python {sys.argv[0]} <<algorithm>, <input filename> <output filename>")
sys.exit(1)
# Extract arguments
algo = sys.argv[1]
input_filename = sys.argv[2]
output_filename = sys.argv[3]
# Call the function with command-line arguments
Live Share
Ln 10, Col 32
Spaces: 4
UTF-8
CRLF
Python
3.12.2 64-bit
Go Live
Search
ENG
US
3:07 AM
6/5/2024


C:\Windows\System32\cmd.e
D:\MMH\extract>python hashes.py
Usage: python hashes.py <<algorithm>, <input filename> <output filename>
D:\MMH\extract>python hashes.py SHA256 22521050.pdf out1.txt
Hashed output save to out1.txt
D:\MMH\extract>cat out1.txt
b88ab5359d2f52203bf6cb9f00ff327e70e71f69f2446f1a2923364fbb98631c
D:\MMH\extract>hashes.exe SHA256 22521050.pdf out2.txt
Hashed saved to out2.txt
D:\MMH\extract>cat out2.txt
b88ab5359d2f52203bf6cb9f00ff327e70e71f69f2446f1a2923364fbb98631c
D:\MMH\extract>
Search
ENG
US
3:07 AM
6/5/2024