

```
File Edit Selection View Go Run ... extract
hashes.py X
hashes.py > ...
13 # All of them are strings (char*)
14 hashes = shas.hashes # call hashes function from shas.so;
15 hashes.argtypes = [c_char_p, c_char_p, c_char_p]
16 hashes.restype = None # The function returns void
17
18 # Wrapped functions
19 def wraphashes(algo, input_filename, output_filename):
20     # Convert Python strings to bytes, as ctypes works with bytes
21     algo = algo.encode('utf-8')
22     input_filename = input_filename.encode('utf-8')
23     output_filename = output_filename.encode('utf-8')
24
25
26     # Call the C function
27     hashes(algo, input_filename, output_filename)
28
29 if __name__ == "__main__":
30     # Check if the correct number of arguments have been passed (3 arguments + 1 for the script name)
31     if len(sys.argv) != 4:
32         print(f"Usage: python {sys.argv[0]} <algorithm>, <input filename> <output filename>")
33         sys.exit(1)
34
35     # Extract arguments
36     algo = sys.argv[1]
37     input_filename = sys.argv[2]
38     output_filename = sys.argv[3]
39
40
41     # Call the function with command-line arguments
42     wraphashes(algo, input_filename, output_filename)
43
44 Live Share
Ln 10, Col 32 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit Go Live
```

```
C:\Windows\System32\cmd.e X + v
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>
```

