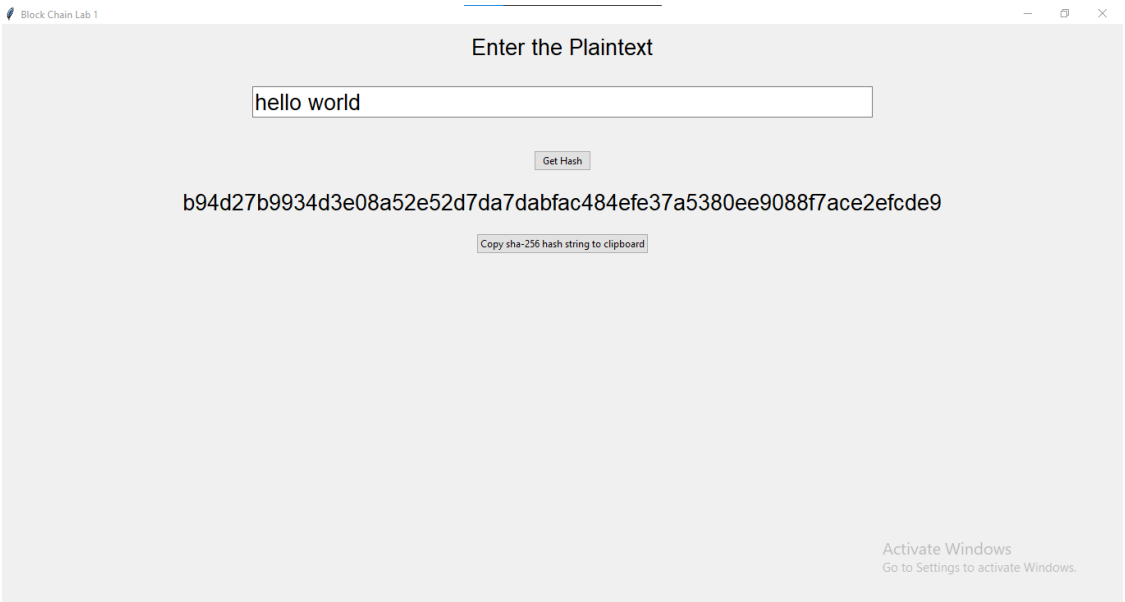
Practical 1

Harshvardhan Singh

1019161

Conversion from plaintext to hash string

```
from tkinter import Tk,ttk
from hashlib import sha256
class crypto:
    def __init__(self) -> None:
        self.plaintext = None
        self.sha_1_hash_string = None
        self.md_5_hash_string = None
    def hash(self,plaintext):
        self.plaintext = plaintext
        self.sha_256_hash_string =
sha256(str(self.plaintext).encode()).hexdigest()
        return self.sha_256_hash_string
class GUI:
    def __init__(self) -> None:
        # setting up basics
        self.window = Tk()
        self.widget = ttk
        self.window.state("zoomed")
        self.window.title("Block Chain Lab 1")
        # putting up widgets
        # input label
        self.request_input_label =
self.widget.Label(text="Enter the
Plaintext",font=("Arial",20))
        self.request_input_label.pack(pady=10)
        self.input_entry =
self.widget.Entry(font=("Arial",20),width=50)
        self.input_entry.pack(pady=20)
        self.ok_button = self.widget.Button(text='Get
Hash',command=self.getHashButton,width=10)
        self.ok_button.pack(pady=20)
        # Result label
        self.results_label =
self.widget.Label(text='',font=("arial",20))
        self.results_label.pack()
        # Copy to clipboard Buttons
        self.copy_sha_hash_button =
self.widget.Button(text='Copy sha-256 hash string to
clipboard',command=self.copy_sha_hash_button)
        self.copy_sha_hash_button.pack(pady=20)
        # get hash button command
    def getHashButton(self):
        self.get_input = str(self.input_entry.get())
        self.hashira = crypto()
        self.results=self.hashira.hash(plaintext=self.get_input)
        self.result_string = f"""{self.results}"""
        self.results_label['text'] =  self.result_string
    # sha 256 copy Button Command
    def copy_sha_hash_button(self):
        print(f"{self.results[0]}")
        self.window.clipboard_clear()
        self.window.clipboard_append(f"{self.results[0]}")
        # the magic begins here
def do_the_magic(self)->None:
        self.window.mainloop()
if __name__ == "__main__":
    GUI().do_the_magic()
```



Implementing the mining process

```
import hashlib
import datetime
chain = []
class Block:
def __init__(self,index,nonce,previous_hash,message):
self.index = index
self.nonce = nonce
self.previous_hash = previous_hash
self.message = message
self.timestamp = str(datetime.datetime.now())
def printChain():
for block in chain:
print("----------")
print("Index: ",block.index)
print("Nonce: ",block.nonce)
print("Previous Hash: ",block.previous_hash)
print("Current Hash: ",getHash(block))
print("Message: ",block.message)
print("Timestamp: ",block.timestamp)
def getHash(block):
return hashlib.sha256(str(block).encode())
def getPreviousHash():
if len(chain)==0:
previous_hash = 0
else:
previous_hash = getHash(chain[-1])
return previous_hash
def validate(block):
blockHash = getHash(block)
for blockchainBlock in chain:
if blockHash==getHash(blockchainBlock):
return False
return True
def addNewBlock(message):
nonce = 1000
if len(chain)==0:
previousNonce = 0
else:
previousNonce = chain[-1].nonce
while True:
nonce = nonce + 1
hashValidator = hashlib.sha256(str(previousNonce**2 -
nonce**2).encode()).hexdigest()
if hashValidator[:4] == '0000':
newBlock =
Block(len(chain)+1,nonce,getPreviousHash(),message)
if validate(newBlock):
chain.append(newBlock)
return
continue
msg = raw_input("Enter you data to be added in the
blockchain: ")
while msg != "-1":
addNewBlock(msg)
printChain()
msg = raw_input("Enter you data to be added in the
blockchain: ")
```

```
admininstrator@CSLAB4-COMP11:~/Documents/53$ python mining.py
Enter you data to be added in the blockchain: Blockchain is a sem 7 subject.
----------
('Index: ', 1)
('Nonce: ', 113932)
('Previous Hash: ', 0)
('Current Hash: ', <sha256 HASH object @ 0x7fc52182eab0>)
('Message: ', 'Blockchain is a sem 7 subject.')
('Timestamp: ', '2022-07-29 12:43:04.916912')
Enter you data to be added in the blockchain: Blockchain is a chain of blocks
----------
('Index: ', 1)
('Nonce: ', 113932)
('Previous Hash: ', 0)
('Current Hash: ', <sha256 HASH object @ 0x7fc52182eba0>)
('Message: ', 'Blockchain is a sem 7 subject.')
('Timestamp: ', '2022-07-29 12:43:04.916912')
----------
('Index: ', 2)
('Nonce: ', 79636)
('Previous Hash: ', <sha256 HASH object @ 0x7fc52182eab0>)
('Current Hash: ', <sha256 HASH object @ 0x7fc52182eba0>)
('Message: ', 'Blockchain is a chain of blocks')
('Timestamp: ', '2022-07-29 12:43:29.566690')
Enter you data to be added in the blockchain: Blockchain lab is lab 504.
----------
('Index: ', 1)
('Nonce: ', 113932)
('Previous Hash: ', 0)
('Current Hash: ', <sha256 HASH object @ 0x7fc52182ebd0>)
('Message: ', 'Blockchain is a sem 7 subject.')
('Timestamp: ', '2022-07-29 12:43:04.916912')
----------
('Index: ', 2)
('Nonce: ', 79636)
('Previous Hash: ', <sha256 HASH object @ 0x7fc52182eab0>)
('Current Hash: ', <sha256 HASH object @ 0x7fc52182ebd0>)
('Message: ', 'Blockchain is a chain of blocks')
('Timestamp: ', '2022-07-29 12:43:29.566690')
----------
('Index: ', 3)
('Nonce: ', 122720)
('Previous Hash: ', <sha256 HASH object @ 0x7fc52182eae0>)
('Current Hash: ', <sha256 HASH object @ 0x7fc52182ebd0>)
('Message: ', 'Blockchain lab is lab 504.')
('Timestamp: ', '2022-07-29 12:44:04.787125')
Enter you data to be added in the blockchain: -1
```