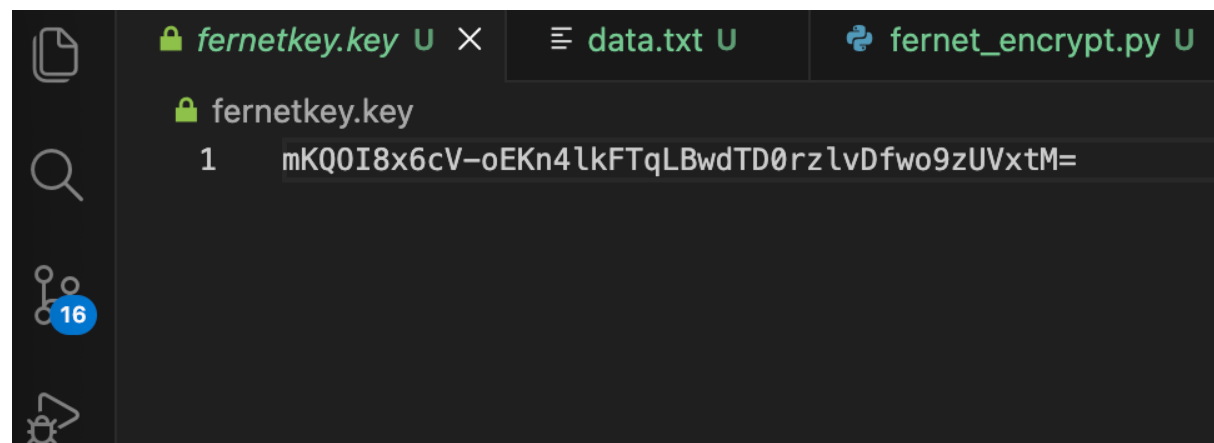


## Task 1:

### Fernet encryption

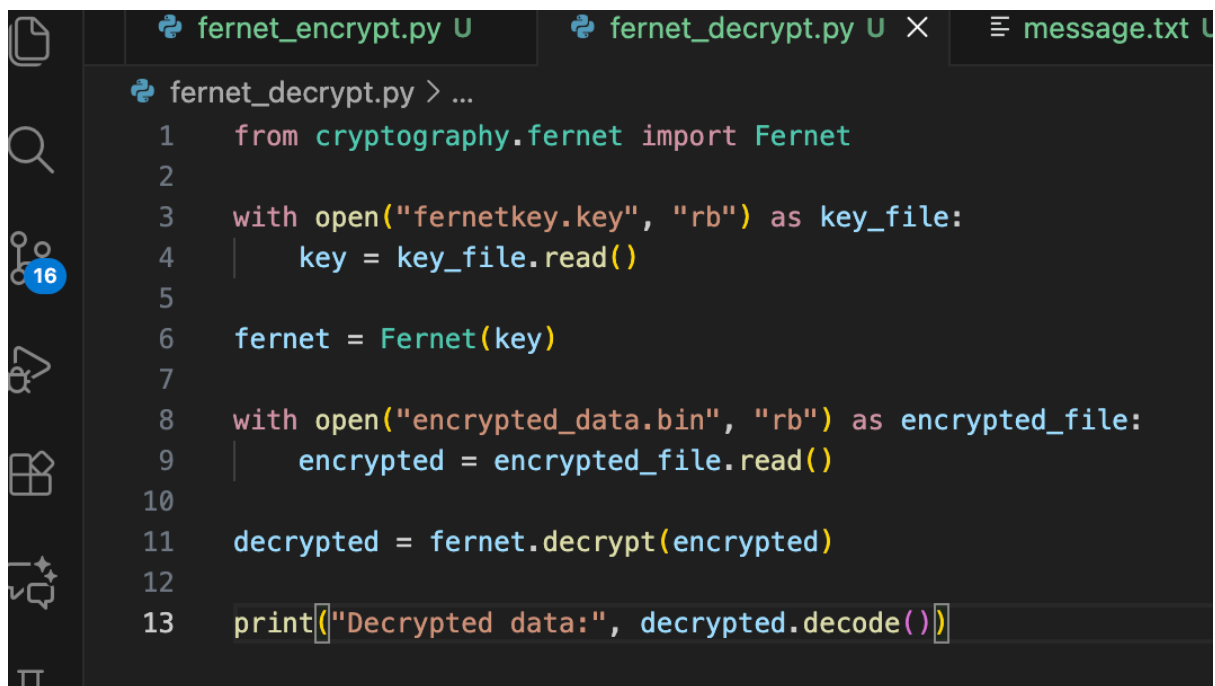
```
fernet_encrypt.py > ...
1  from cryptography.fernet import Fernet
2
3  key = Fernet.generate_key()
4  fernet = Fernet(key)
5
6  with open("fernetkey.key", "wb") as key_file:
7      key_file.write(key)
8
9  with open("data.txt", "rb") as file:
10     original = file.read()
11
12     encrypted = fernet.encrypt(original)
13
14     with open("encrypted_data.bin", "wb") as encrypted_file:
15         encrypted_file.write(encrypted)
16     print("Encrypted data:", encrypted)
17
18
```

### Fernet Key



The screenshot shows a code editor with three tabs: `fernetkey.key`, `data.txt`, and `fernet_encrypt.py`. The `fernetkey.key` tab is active, displaying the generated Fernet key on line 1: `mKQ0I8x6cV-oEKn4lkFTqLBwdTD0rzlvDfwo9zUVxtM=`. The editor interface includes a sidebar with icons for file explorer, search, and a notification badge showing the number 16.

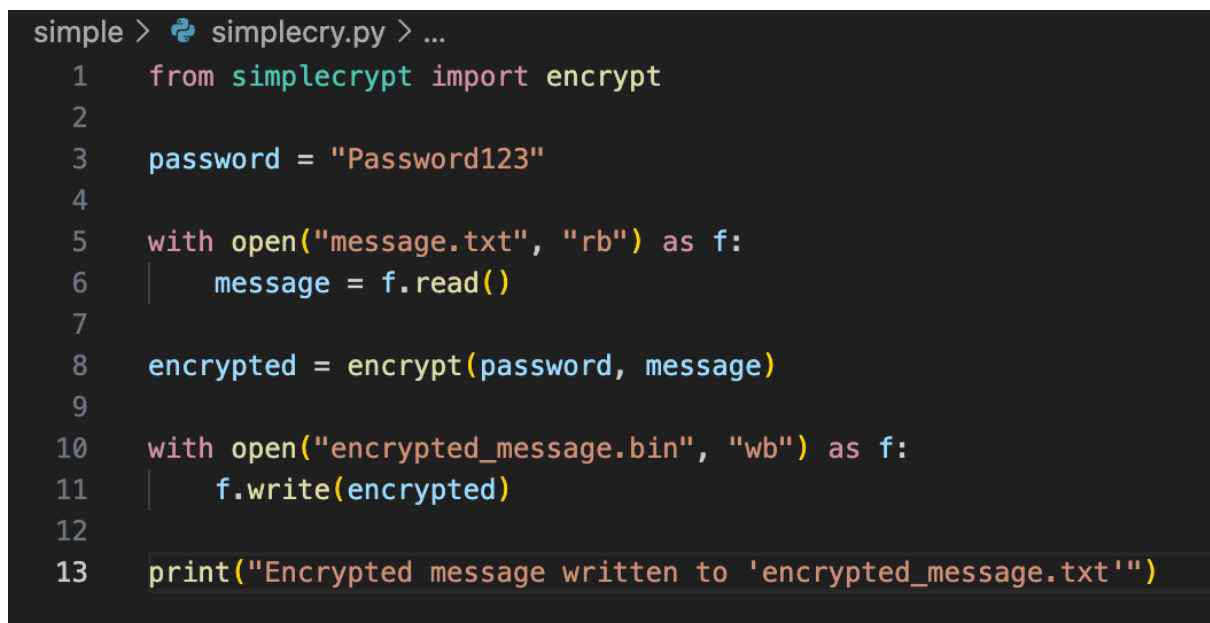
## Fernet Decryption

A screenshot of a code editor with a dark theme. The editor has three tabs at the top: 'fernet\_encrypt.py U', 'fernet\_decrypt.py U X', and 'message.txt U'. The active tab is 'fernet\_decrypt.py U'. The code is written in Python and is as follows:

```
fernet_decrypt.py > ...
1  from cryptography.fernet import Fernet
2
3  with open("fernetkey.key", "rb") as key_file:
4      key = key_file.read()
5
6  fernet = Fernet(key)
7
8  with open("encrypted_data.bin", "rb") as encrypted_file:
9      encrypted = encrypted_file.read()
10
11  decrypted = fernet.decrypt(encrypted)
12
13  print("Decrypted data:", decrypted.decode())
```

## Task 2:

### encryption

A screenshot of a code editor with a dark theme. The editor has one tab at the top: 'simplecry.py > ...'. The code is written in Python and is as follows:

```
simple > simplecry.py > ...
1  from simplecrypt import encrypt
2
3  password = "Password123"
4
5  with open("message.txt", "rb") as f:
6      message = f.read()
7
8  encrypted = encrypt(password, message)
9
10 with open("encrypted_message.bin", "wb") as f:
11     f.write(encrypted)
12
13 print("Encrypted message written to 'encrypted_message.txt'")
```

## Decryption

```
simple > 🐡 desimple.py > ...
1  from simplecrypt import decrypt
2
3  password = "Password123"
4
5  with open("encrypted_message.bin", "rb") as f:
6      encrypted = f.read()
7
8  decrypted = decrypt(password, encrypted)
9
10 print("Decrypted message:", decrypted.decode())
```

## Task 3:

```
simple > 🐡 script.py > ...
1  import paramiko
2
3  hostname = "localhost"
4  port = 2222
5  username = "admin"
6  password = "root"
7
8  with open("command.txt", "r") as f:
9      command = f.read().strip()
10
11  ssh = paramiko.SSHClient()
12  ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
13
14  ssh.connect(hostname, port=port, username=username, password=password)
15
16  stdin, stdout, stderr = ssh.exec_command(command)
17
18  output = stdout.read().decode()
19  with open("ssh_output.txt", "w") as f:
20      f.write(output)
21
22  ssh.close()
23  print("Done! Check ssh_output.txt")
24
```