Esercizi python

```python
def acquisisci__parole():
    lista_parole = input(" Inserisci 3 parole( separate da spazi): ")
    parole = lista_parole.split()
    return parole

def calcola_lunghezza(parole):
    lunghezze = []
    for parola in parole:
          lunghezze.append(len(parola))
    return lunghezze

presa_parole = acquisisci__parole()
lunghezza_parole = calcola_lunghezza(presa_parole)
print (lunghezza_parole)
```

```
┌──(nicholas㉿kali)-[~/Scrivania]
└─$ python eserciziopython.py
 Inserisci 3 parole( separate da spazi): ciao voglia giuseppe
[4, 6, 8]
```

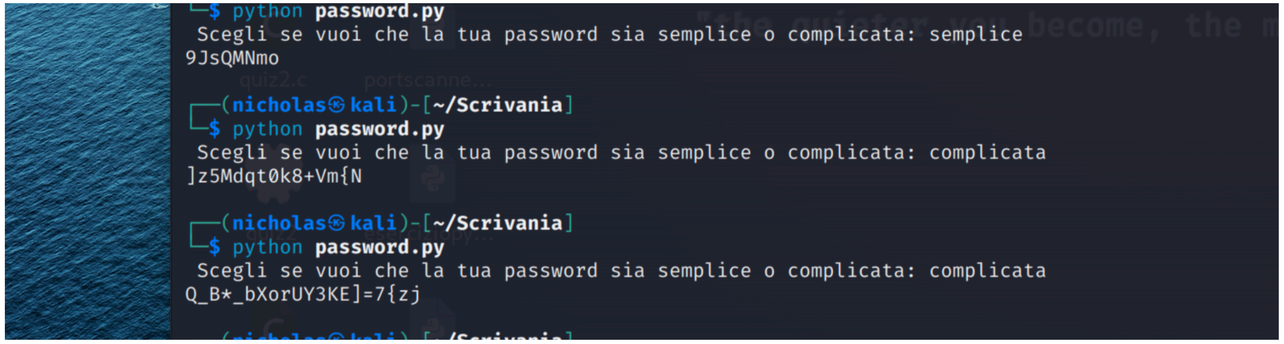
- In questa slide viene mostrato l'esercizio relativo alla conversione di parole quantificando in numeri le lettere presenti all'interno di ciascuna parola

```
 GNU nano 7.2                                                password.py
import random
import string

def genera_password(scelta):
    if scelta == "semplice":
        lunghezza = 8
        caratteri = string.ascii_letters + string.digits
    elif scelta == "complicata":
        lunghezza = 20
        caratteri = string.printable.replace(" ", "").replace("\t", "").replace("\n", "")
    else:
        return "Scelta non valida. Si prega di selezionare 'semplice' o 'complicata'."

    password = ''.join(random.choice(caratteri) for _ in range(lunghezza))
    return password

scelta_password = genera_password
scelta  = input (" Scegli se vuoi che la tua password sia semplice o complicata: ")
print (scelta_password(scelta))
```



- In quest' altra slide viene mostrata l'esecuzione dell' esercizio di generazione di password semplice o complicata

```
  File "/home/nicholas/Scrivania/provasocket.py", line 10, in <module>
    connection, address = s.accept()
                          ^^^^^^^^^^
  File "/usr/lib/python3.11/socket.py", line 294, in accept
    fd, addr = self._accept()
               ^^^^^^^^^^^^^^
KeyboardInterrupt


┌──(nicholas㉿kali)-[~/Scrivania]
└─$ nano provasocket.py

┌──(nicholas㉿kali)-[~/Scrivania]
└─$ python provasocket.py
 Server started! waiting for connection...
 Client connected with address:  ('192.168.50.110', 35994)
```

```
Port  35989 - CLOSED
Port  35990 - CLOSED
Port  35991 - CLOSED
Port  35992 - CLOSED
Port  35993 - CLOSED
Port  35994 - CLOSED
Port  35995 - CLOSED
Port  35996 - CLOSED
Port  35997 - CLOSED
Port  35998 - CLOSED
Port  35999 - CLOSED

┌──(nicholas㉿kali)-[~/Scrivania]
└─$ python portscanner.py
 Enter the IP addres to scan: 192.168.50.110
 Enter the port range to scan (es 5-200): 44441-44449
Scanning host  192.168.50.110 from port  44441 to port  44449
Port  44441 - CLOSED
Port  44442 - CLOSED
Port  44443 - CLOSED
 *** Port  44444 - OPEN ***
Port  44445 - CLOSED
Port  44446 - CLOSED
Port  44447 - CLOSED
Port  44448 - CLOSED

┌──(nicholas㉿kali)-[~/Scrivania]
└─$
```

```
Parallel DNS resolution of 1 host. Timing: About 0

┌──(nicholas㉿kali)-[~]
└─$ netcat 192.168.50.110 44444

┌──(nicholas㉿kali)-[~]
```

- Infine riporto l’esercizio relativo alle slide su python in modo da osservare che utilizzando netcat per connettersi al server creato da noi, è possibile attraverso il programma di portscanner individuare la porta su cui ci si è connessi