

## ESERCIZIO W7D1 PARTE 2

La seconda traccia è la seguente:

- Scrivi una funzione generatrice di password. La funzione deve generare una stringa alfanumerica di 8 caratteri qualora l'utente voglia una password semplice, o di 20 caratteri ascii qualora desideri una password più complicata.

```
import random
import string

def generatore_password(x):
    if x==True:
        lunghezza = 20
    else:
        lunghezza = 8

    caratteri = string.ascii_letters + string.digits

    # all'interno di caratteri inseriamo tutte le lettere dell'alfabeto maiuscole e minuscole
    # assieme a tutti i numeri da 0 a 9

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

    # nella variabile password uniamo in una stringa le lettere contenute in caratteri, prese
    # casualmente grazie all'oggetto .choice. Il ciclo for ci darà la lunghezza adatta
    return password

scelta = input("Desideri una password complessa? (Si/No): ").lower()

if scelta == "si":
    password_generata = generatore_password(True)
elif scelta == "no":
    password_generata = generatore_password(False)
else:
    print("Quello che hai scritto non mi sembra ne un no ne un si")
    exit()
print("La tua password generata è:", password_generata)

# Dopo la funzione troviamo finalmente la scelta, tra si o no. if darà True per la funzione
# mentre elif False. Nel caso venga scritto altro, else si occuperà della casualità
```

Programma Python

```
(kali㉿kali)-[~/Python]
$ nano pass_generator.py

(kali㉿kali)-[~/Python]
$ python pass_generator.py
Desideri una password complessa? (Si/No): si
La tua password generata è: jdyKoumaJPPXrxDuVllm

(kali㉿kali)-[~/Python]
$ python pass_generator.py
Desideri una password complessa? (Si/No): no
La tua password generata è: gAQ4ov6H

(kali㉿kali)-[~/Python]
$ █ Home
```

Tutte le risposte  
ed il loro output

```
(kali㉿kali)-[~/Python]
$ python pass_generator.py
Desideri una password complessa? (Si/No): xx
Quello che hai scritto non mi sembra ne un no ne un si
```

risposta sbagliata