Consegna:

EPICODE

Esercizio
Python per Hacker Pt. 2

Traccia:

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.

Soluzione:

```
3
4  lunghezza = int(input("Quanti caratteri vuoi nella tua password? 8 o 20? "))
5  numeri = input("Vuoi che la tua password contenga numeri? Digita 'yes' o 'no': ")
6  simboli = input("Vuoi che la tua password contenga simboli? Digita 'yes' o 'no': ")
7  lettere = input("Vuoi che la password contenga lettere? Digita 'yes' o 'no': ")
8
9  def generatore_password(lunghezza, numeri, simboli, lettere):
10     caratteri = ""
11
12     if numeri == "yes":
13         caratteri += string.digits
14
15     if simboli == "yes":
16         caratteri += "?!^_@$%"
17
18     if lettere == "yes":
19         caratteri += string.ascii_letters
Ln: 5, Col: 82
```

```
10     caratteri = ""
11
12     if numeri == "yes":
13         caratteri += string.digits
14
15     if simboli == "yes":
16         caratteri += "?!^_@$%"
17
18     if lettere == "yes":
19         caratteri += string.ascii_letters
20
21     lunghezza = min(lunghezza, len(caratteri))
22     password = ''.join(random.choice(caratteri) for _ in range(lunghezza))
23
24     return password
25
26 print(generatore_password(lunghezza, numeri, simboli, lettere))
Ln: 5, Col: 82
```

```
Quanti caratteri vuoi nella tua password? 8 o 20?
8
Vuoi che la tua password contenga numeri? Digita 'yes' o 'no':
yes
Vuoi che la tua password contenga simboli? Digita 'yes' o 'no':
yes
Vuoi che la password contenga lettere? Digita 'yes' o 'no':
yes
^hq3D6rZ


** Process exited - Return Code: 0 **
```

1. Richiesta dell'utente per la lunghezza e le opzioni della password:

lunghezza = int(input("Quanti caratteri vuoi nella tua password? 8 o 20? ")) ecc...

- L'utente fornisce la lunghezza desiderata e le preferenze per numeri, simboli e lettere nella password.

- `input` viene utilizzato per ottenere l'input dell'utente.

2. Definizione della funzione `generatore_password`:

def generatore_password(lunghezza, numeri, simboli, lettere):

caratteri = ""

- Viene definita una funzione `generatore_password` che accetta la lunghezza e le opzioni per includere numeri, simboli e lettere nella password.

- La variabile `caratteri` viene inizializzata come una stringa vuota per contenere i caratteri possibili per la password.

3. Aggiunta di numeri, simboli e lettere alla stringa `caratteri:

if numeri == "yes":

caratteri += string.digits

if simboli == "yes":

caratteri += "?!^_@$%"

if lettere == "yes":

caratteri += string.ascii_letters

- Se l'utente vuole numeri, li aggiungiamo alla stringa di caratteri.

- Se l'utente vuole simboli, li aggiungiamo alla stringa di caratteri.

- Se l'utente vuole lettere, le aggiungiamo alla stringa di caratteri.

4. Limitazione della lunghezza massima alla lunghezza disponibile:

lunghezza = min(lunghezza, len(caratteri))

- Per evitare che la lunghezza della password superi la lunghezza effettiva dei caratteri disponibili, utilizziamo `min` per selezionare il valore più piccolo tra la lunghezza desiderata e la lunghezza effettiva della stringa `caratteri`.

5. Generazione della password e restituzione del risultato:

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

return password

- Utilizziamo una comprensione di lista per generare casualmente caratteri dalla stringa `caratteri` per la lunghezza desiderata.

- Uniamo questi caratteri in una stringa usando `join` e restituiamo la password risultante.

6. Chiamata della funzione e stampa della password generata:

print(generatore_password(lunghezza, numeri, simboli, lettere))

- Chiamiamo la funzione `generatore_password` con gli input forniti dall'utente e stampiamo la password generata.