**Inputs:**

1. **Number of Passwords:**
   * The user is prompted to enter the number of passwords they want to generate.

num\_passwords = get\_valid\_input("How many passwords do you want to generate? ", min\_value=1)

1. **Length of Each Password:**
   * For each password, the user is prompted to enter the desired length. The minimum length is enforced as 3.

for i in range(num\_passwords):

length = get\_valid\_input(f"Enter the length of Password #{i + 1} (minimum length is 3): ")

password\_lengths.append(length)

**Outputs:**

1. **Generated Passwords:**
   * The program prints each generated password, one per line.

for i, password in enumerate(passwords, start=1):

print(f"Password #{i} = {password}")

**Example Scenario:**

1. **User Input:**
   * Number of passwords to generate: 3
   * Lengths of the passwords: 5, 8, 2 (the last length will be adjusted to 3)
2. **Program Output:**
   * The program generates and prints three passwords.

**Example Execution:**

How many passwords do you want to generate? 3

Generating 3 passwords

Enter the length of Password #1 (minimum length is 3): 5

Enter the length of Password #2 (minimum length is 3): 8

Enter the length of Password #3 (minimum length is 3): 2

Value should be at least 3. Setting to 3.

Password #1 = hG2$a

Password #2 = rP1&W8xQ

Password #3 = fJ3

**Detailed Breakdown:**

1. **Input Prompt 1:**

How many passwords do you want to generate?

* + The user inputs 3.

1. **Input Prompt 2:**

Enter the length of Password #1 (minimum length is 3):

* + The user inputs 5.

1. **Input Prompt 3:**

Enter the length of Password #2 (minimum length is 3):

* + The user inputs 8.

1. **Input Prompt 4:**

Enter the length of Password #3 (minimum length is 3):

* + The user inputs 2.
  + The program adjusts this to 3 and informs the user.

Value should be at least 3. Setting to 3.

1. **Output:**

Password #1 = hG2$a

Password #2 = rP1&W8xQ

Password #3 = fJ3

**Summary:**

* **Input Handling:**
  + The user is prompted for the number of passwords and their lengths.
  + Input validation ensures the minimum length is 3 and handles invalid inputs gracefully.
* **Output:**
  + The program generates complex passwords using lowercase, uppercase letters, digits, and punctuation.
  + Each password is printed to the console with its corresponding number.