1. Password Generator

1. Import required libraries: random and string.

2. Define password length.

3. Create a character pool (uppercase, lowercase, digits, symbols).

4. Randomly select characters from the pool.

5. Generate and display the password.

```python
import random
import string

def generate_password(length=12):
    characters = string.ascii_letters + string.digits + string.punctuation
    password = ''.join(random.choice(characters) for _ in range(length))
    return password

print("Generated Passsword:",generate_password(12))
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\Administrator\passwordgenerator> python password_generator.py
Generated Passsword: lX/}{b<wd<$p
PS C:\Users\Administrator\passwordgenerator>
```

2. To-Do List (CLI)

1. Create a list to store tasks.

2. Provide options: Add, View, Remove, Exit.

3. Loop until the user exits.

```python
tasks = []

while True:
    print("\n1. Add Task\n2. View Tasks\n3. Remove Task\n4. Exit")
    choice = input("Enter choice: ")

    if choice == "1":
        task = input("Enter task: ")
        tasks.append(task)
        print("Task added!")

    elif choice == "2":
        print("\nTo-Do List:")
        for idx, task in enumerate(tasks, 1):
            print(f"{idx}. {task}")

    elif choice == "3":
        task_num = int(input("Enter task number to remove: "))
        if 0 < task_num <= len(tasks):
            tasks.pop(task_num - 1)
            print("Task removed!")
        else:
            print("Invalid task number.")

    elif choice == "4":
        break

    else:
        print("Invalid choice. Try again.")
```

Output:

```
Enter choice: 1
Enter task: Drink more water
Task added!

1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 1
Enter task: Sleep well
Task added!

1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 2

To-Do List:
1. Hit the gym
2. Eat healthy food
3. Drink more water
4. Sleep well

1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 3
Enter task number to remove: Sleep well
```

3. Weather App (API-based)

1. Sign up for OpenWeatherMap API and get an API key.

2. Use requests to fetch weather data.

3. Display temperature, weather condition, and city name.

```python
import requests

API_KEY = "cdbe8d92a6ad557a042bd8c83f8786e1"  # Get from https://openweathermap.org/api
city = input("Enter city name: ")

# Construct the URL properly
url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API_KEY}&units=metric"

response = requests.get(url).json()

if response["cod"] == 200:
    print(f"City: {response['name']}")
    print(f"Temperature: {response['main']['temp']}°C")
    print(f"Weather: {response['weather'][0]['description']}")
else:
    print("City not found!")
```

Output:

```
PS C:\Users\Administrator\passwordgenerator> python weather.py
Enter city name: paris
City: Paris
Temperature: 4.64°C
Weather: overcast clouds
PS C:\Users\Administrator\passwordgenerator>
```

4. Number Guessing Game

1. Generate a random number between 1-100.

2. Ask the user to guess.

3. Give hints if the guess is too high/low.

4. Continue until guessed correctly.

num_guess.py > ...

```python
import random

# Generate a random number between 1 and 100
number = random.randint(1, 100)

while True:
    guess = int(input("Guess the number (1-100): "))

    if guess < number:
        print("Too low! Try again.")
    elif guess > number:
        print("Too high! Try again.")
    else:
        print("Congratulations! You guessed it right.")
        break
```

Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell

```
Guess the number (1-100): 35
Too high! Try again.
Guess the number (1-100): 32
Too high! Try again.
Guess the number (1-100): 31
Congratulations! You guessed it right.
PS C:\Users\Administrator\passwordgenerator>
```

5. QR Code Generator

1. Install qrcode library (pip install qrcode).

2. Take user input (text/link) to convert.

3. Generate and save the QR code.

```python
import qrcode

# Get user input for the QR code data
data = input("Enter text or URL: ")

# Create the QR code
qr = qrcode.make(data)

# Save the QR code as a PNG file
qr.save("qrcode.png")

# Confirm that the QR code has been generated and saved
print("QR Code generated and saved as 'qrcode.png'!")
```

Output:

```
PS C:\Users\Administrator\passwordgenerator> python qr.py
Enter text or URL: www.google.com
QR Code generated and saved as 'qrcode.png'!
PS C:\Users\Administrator\passwordgenerator>
```