Name: **Date:** **Class:**

Advanced Computational Activities

The following three activities—coding design, implementation, and debugging—provide a glimpse into how software engineers work. These tasks are based on real-world problems from materials science, offering students insight into the practical application of programming in scientific research.

Activity 1: Simulate a Transistor as a Digital Switch in Python

Time: 90 minutes
Level: Computer Science I, AP Computer Science Principles

Challenge Prompt:
Imagine you are designing a digital circuit for a smart device. You will use a transistor to control when electricity flows, just like in real computers. Your goal is to simulate how an NPN transistor works like a switch in Python. The transistor will turn ON (conduct current) if the base voltage is high enough, and otherwise.

```
def simulate_transistor(V_base, V_collector, V_emitter, V_threshold=0.7):
    # TODO: Calculate base-emitter voltage

    # TODO: Use if-statement to check if transistor is ON
           #Transistor is ON:  current can flow

           #Transistor is OFF: no current flows

    # TODO: Print output voltage

# TODO: Example tests:
```

Bonus Challenges:
- Simulate a NOT gate: input 0V → output 5V; input 5V → output 0V.
- Turn it into an AND or NAND gate using 2 transistors.
- Add a graph of input vs. output voltage using matplotlib.
- Show transistor behavior in analog mode (for advanced students).

Name: **Date:** **Class:**

Activity 2. Simulate a Smart Memristor in Python

Time: 90 minutes
Level: Computer Science I, AP Computer Science Principles

Challenge Prompt: You are an engineer designing a memory device of the future. Your task is to simulate a memristor, a special electronic component that remembers how much current passed through it. Use Python to create a program that:

- Takes a current input over time (e.g., a list or sine wave).
- Tracks how the resistance changes based on the input.
- Calculates and plots the voltage over time.

Starter Code:

```
import matplotlib.pyplot as plt

# Create a simple list of input currents (positive and negative)
I = [0.001, 0.002, 0.001, -0.001, -0.002, -0.001, 0.0, 0.001, 0.002]
time = list(range(len(I)))

# Starter values
R_on = 100
R_off = 10000
w = 0.5  # memory state (0 to 1)

# Empty lists to store output
R_values = []
V_values = []

for current in I:
    # TODO: update w based on current
    # TODO: calculate resistance based on w
    # TODO: calculate voltage = current * resistance
    # TODO: append to R_values and V_values
    pass

# TODO: Plot voltage over time
```

Bonus Challenges:

- Plot resistance vs. time.
- Change the input current to a sine wave.
- Plot a hysteresis curve (Voltage vs Current).
- Show how the memristor could store binary data (if resistance is low → 1, high → 0).

Name: **Date:** **Class:**

Activity 3: Simulate How a Missing Atom Changes a Crystal's Energy in Python

Time: 2 - 3 90 minute-sessions
Level: Computer Science III or IB HL

Scenario:
You are working as a scientist on next-generation solar cells. You need to understand how missing atoms (defects) affect the energy of a material. In real physics, scientists use Density Functional Theory (DFT) to calculate how electrons and atoms behave. Today, you will design your own simplified simulation inspired by DFT!

Your Task:
Design a Python program that:

- Creates a 1D list of atoms spaced evenly on a line.
- Writes a function to calculate the total energy of the system. (Hint: Use an equation where energy depends on distance between atoms; the closer they are, the stronger the effect.)
- Removes one atom from the middle to simulate a defect.
- Recalculates the total energy and compares it to the perfect crystal.
- Uses a scatter plot to visualize the atoms before and after the defect.

Tips for Planning:
Before you start coding, answer the following:

- What kind of data structure will hold atom positions?
- What kind of loop will you use to calculate the energy of the system?
- What does it mean for the system to be "stable"?

Bonus (Optional):

- Try removing two atoms. How does energy change?
- Plot energy before and after defect using matplotlib.
- Let atoms "move" to reduce energy (like real materials do)!

Question: If a regular computer can simulate 10 atoms in 1 minute, how many atoms could it simulate in 1 hour? What if we needed to simulate 1 billion atoms? What options do we have? (Encourages them to think about scale and the limits of traditional computing.)