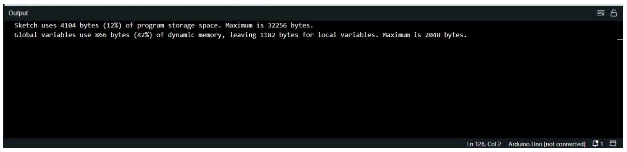
```
// Example code for Arduino Uno
#include <SoftwareSerial.h>

// Define Bluetooth module communication pins (RX, TX)
SoftwareSerial BTSerial(10, 11); // RX (Pin 10) and TX (Pin 11)

// Define relay control pins for multiple devices
int relayPin1 = 8;  // Relay 1 to control Device 1 (e.g., Lamp)
int relayPin2 = 9;  // Relay 2 to control Device 2 (e.g., Fan)
int relayPin3 = 7;  // Relay 3 to control Device 3 (e.g., Air Conditioner)
int relayPin4 = 6;  // Relay 4 to control Device 4 (e.g., Heater)

// Variables to store Bluetooth command input
char incomingCommand;
boolean isDevice1On = false;
boolean isDevice2On = false;
boolean isDevice3On = false;
boolean isDevice4On = false;

// Setup function - Initializes pins and starts Bluetooth communication
void setup() {
  // Start serial communication for debugging and monitoring
  Serial.begin(9600);
  BTSerial.begin(9600);  // Initialize Bluetooth communication

  // Set relay pins as OUTPUTs to control the devices
  pinMode(relayPin1, OUTPUT);
  pinMode(relayPin2, OUTPUT);
  pinMode(relayPin3, OUTPUT);
  pinMode(relayPin4, OUTPUT);

  // Initially, turn off all devices (set relays to LOW)
  digitalWrite(relayPin1, LOW);
  digitalWrite(relayPin2, LOW);
  digitalWrite(relayPin3, LOW);
  digitalWrite(relayPin4, LOW);

  // Output message to indicate the system is ready
  Serial.println("Home Automation System Ready. Awaiting Bluetooth Commands...");
  Serial.println("Commands: 1=Device1 ON, 0=Device1 OFF, 3=Device2 ON, 2=Device2 OFF,
5=Device3 ON, 4=Device3 OFF, 7=Device4 ON, 6=Device4 OFF");
}

// Main loop function - Continuously checks for Bluetooth commands
```

```
void loop() {
  // Check if data is available from the Bluetooth module
  if (BTSerial.available()) {
    incomingCommand = BTSerial.read();  // Read the incoming Bluetooth command

    // Display the received command for debugging purposes
    Serial.print("Received Command: ");
    Serial.println(incomingCommand);

    // Control Device 1 (e.g., Lamp)
    if (incomingCommand == '1') {
      digitalWrite(relayPin1, HIGH);  // Turn on Device 1 (Relay 1)
      isDevice1On = true;
      Serial.println("Device 1 (Lamp) is ON");
    }
    else if (incomingCommand == '0') {
      digitalWrite(relayPin1, LOW);  // Turn off Device 1 (Relay 1)
      isDevice1On = false;
      Serial.println("Device 1 (Lamp) is OFF");
    }

    // Control Device 2 (e.g., Fan)
    else if (incomingCommand == '3') {
      digitalWrite(relayPin2, HIGH);  // Turn on Device 2 (Relay 2)
      isDevice2On = true;
      Serial.println("Device 2 (Fan) is ON");
    }
    else if (incomingCommand == '2') {
      digitalWrite(relayPin2, LOW);  // Turn off Device 2 (Relay 2)
      isDevice2On = false;
      Serial.println("Device 2 (Fan) is OFF");
    }

    // Control Device 3 (e.g., Air Conditioner)
    else if (incomingCommand == '5') {
      digitalWrite(relayPin3, HIGH);  // Turn on Device 3 (Relay 3)
      isDevice3On = true;
      Serial.println("Device 3 (Air Conditioner) is ON");
    }
    else if (incomingCommand == '4') {
      digitalWrite(relayPin3, LOW);  // Turn off Device 3 (Relay 3)
      isDevice3On = false;
      Serial.println("Device 3 (Air Conditioner) is OFF");
    }
    // Control Device 4 (e.g., Heater)
    else if (incomingCommand == '7') {
      digitalWrite(relayPin4, HIGH);  // Turn on Device 4 (Relay 4)
      isDevice4On = true;
      Serial.println("Device 4 (Heater) is ON");
    }
    else if (incomingCommand == '6') {
      digitalWrite(relayPin4, LOW);  // Turn off Device 4 (Relay 4)
      isDevice4On = false;
      Serial.println("Device 4 (Heater) is OFF");
    }

    // Handle invalid commands
    else {
      Serial.println("Invalid command received. Please send a valid command.");
    }

    // Update the state of the devices (for debugging or future enhancements)
    displayDeviceStates();
  }

  // Optional: Additional logic can be added here, such as timed actions, or check for Bluetooth signal loss
}

// Function to display the current states of the devices for debugging
void displayDeviceStates() {
  Serial.print("Device 1 (Lamp) is ");
  Serial.println(isDevice1On ? "ON" : "OFF");

  Serial.print("Device 2 (Fan) is ");
  Serial.println(isDevice2On ? "ON" : "OFF");

  Serial.print("Device 3 (Air Conditioner) is ");
  Serial.println(isDevice3On ? "ON" : "OFF");

  Serial.print("Device 4 (Heater) is ");
  Serial.println(isDevice4On ? "ON" : "OFF");
}
```



```
Sketch uses 4104 bytes (12%) of program storage space. Maximum is 32256 bytes.
Global variables use 866 bytes (42%) of dynamic memory, leaving 1182 bytes for local variables. Maximum is 2048 bytes.
```