INF211

Algorithms and Programming I

PROJECT 1 - Student Brief

"4 Band Resistor Decoder"

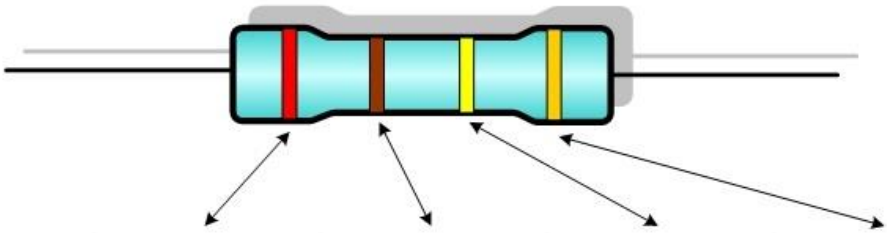
Fall 2025

Write a Python program that reads a 4-band resistor color code in UPPERCASE (format: D1-D2-MULT-TOL) and prints its nominal value (ohms), tolerance (±%), and minimum/maximum values. Use the table in Figure 1 for color-value matching.

A 4-band resistor encodes: Band1 = first digit, Band2 = second digit, Band3 = multiplier (10^n), Band4 = tolerance (±%).

Example: YELLOW–VIOLET–RED–GOLD → 4,7,×10²,±5% → Value = 47×100 = 4700 Ω.

4-band Resistor



Color	1st band value	2nd band value	Multiplier	Tolerances
Black	0	0	× 1	
Brown	1	1	× 10	± 1%
Red	2	2	× 100	± 2%
Orange	3	3	× 1000	± 3%
Yellow	4	4	× 10,000	± 4%
Green	5	5	× 100,000	± 0.5%
Blue	6	6	× 1,000,000	± 0.25%
Violet	7	7	× 10,000,000	± 0.10%
Grey	8	8	× 100,000,000	± 0.05%
White	9	9	× 1,000,000,000	
Gold			× 0.1	± 5%
Silver			× 0.01	± 10%

Figure 1. 4-Band Resistor Color Chart (read from left to right)

The color names are not case-sensitive (e.g., red, Red, RED are all the same). Accepted names are: BLACK, BROWN, RED, ORANGE, YELLOW, GREEN, BLUE, VIOLET, GREY/GRAY, WHITE, GOLD, SILVER.

Rules:

• No external libraries.

• Prohibited: Built-in text or list helper functions such as split, find, count, append, ord, chr, sum, sorted, replace, math, etc.

• Allowed: input, print, len, range, int, float, slicing (s[i:j], s[::-1]), etc.

• Input must be uppercase and in the format D1-D2-MULT-TOL.

• Manual parsing only (no .split() allowed).

• Follow the required function structure exactly.

• If the color bands are entered in reverse order (starting with the tolerance color), your program must detect this and print: "WRONG ORDER, TURN THE RESISTOR AROUND" before computing the correct value.
• If an invalid color name is entered: Invalid color name
• If a missing/irregular number of bands is entered: Invalid format or missing colors
• The first band cannot be BLACK (a resistance starting with a value of 0 is not coded).

Required Function Structure:

Use exactly these functions; only main() may use input() or print():

- **parse_four_bands(line)**
  Read input character by character. On - (ASCII hyphen), one color ends. Return four strings (band1, band2, mult_color, tol_color).

- **color_to_value(color)**
  Map color to a digit (0–9). BLACK=0, BROWN=1, …, WHITE=9.

- **color_to_multiplier_and_tolerance(mult_color, tol_color)**

  Return (exp, tol), where exp is the base-10 exponent (an integer; can be negative, e.g., GOLD → exp = −1) and tol is the tolerance in percent

- **is_upper_letter(ch)**
  Return True if ch is an uppercase letter A..Z

- **def compute_resistor_value(d1, d2, exp, tol):**

  Compute R = $(10 \times d_1 + d_2) \times 10^{exp}$; Rmin, Rmax accordingly.

- **main()**
  Read input, call functions, print four lines exactly as shown below.

```
>>> Input: YELLOW-VIOLET-RED-GOLD
        Value: 4700.00 Ω
        Tolerance: ±5.00%
        Min: 4465.00 Ω
        Max: 4935.00 Ω
>>> Input: BROWN-BLACK-ORANGE-RED
        Value: 10000.00 Ω
        Tolerance: ±2.00%
        Min: 9800.00 Ω
        Max: 10200.00 Ω
>>> Input: ORANGE-ORANGE-BROWN-BROWN
        Value: 330.00 Ω
        Tolerance: ±1.00%
        Min: 326.70 Ω
        Max: 333.30 Ω
>>> Input: YELLOW-PINK-RED-GOLD
        Invalid color code!

>>> Input: yellow-violet-red-gold
        Invalid input! Please use uppercase letters (A–Z).

>>> Input: GREEN-BLUE-GOLD-SILVER
        Value: 5.60 Ω
        Tolerance: ±10.00%
        Min: 5.04 Ω
        Max: 6.16 Ω
>>> Input: GOLD-RED-VIOLET-YELLOW
        WRONG ORDER, TURN THE RESISTOR AROUND
        Value: 4700.00 Ω
        Tolerance: ±5.00%
        Min: 4465.00 Ω
        Max: 4935.00 Ω
>>> Input: BROWN-BLACK-ORANGE-WHITE
        WRONG ORDER, TURN THE RESISTOR AROUND
        Value: 93.00 Ω
        Tolerance: ±1.00%
        Min: 92.07 Ω
        Max: 93.93 Ω
>>> Input: YELLOW-VIOLET-RED-GOLD-
        Invalid format or missing colors!
>>> Input: RED-GREEN-BLUE-YELLOW-BLACK
        Invalid format or missing colors!
>>> Input: BLACK-RED-VIOLET-YELLOW
        Invalid color code!
```