1. Even or Odd

Write a program that asks the user for a number and prints whether it is even or odd.

2. Temperature Converter

Create a program that converts temperatures between Fahrenheit and Celsius. Ask the user which conversion they want to perform ("F to C" or "C to F") and then perform the selected conversion.

3. Grade Classifier

Write a program that takes a numerical grade (0-100) and outputs the corresponding letter grade:

- A: 90-100
- B: 80-89
- C: 70-79
- D: 60-69
- F: 0-59

4. Simple Calculator

Create a calculator that performs basic operations (addition, subtraction, multiplication, division) based on user input. Ask for two numbers and the operation to perform.

5. Age Group Classifier

Write a program that asks for a person's age and classifies them as:

- Child (0-12)
- Teenager (13-19)
- Adult (20-64)
- Senior (65+)

6. Leap Year Checker

Create a program that determines if a given year is a leap year. (A leap year is divisible by 4, except for century years which must be divisible by 400).

7. Positive, Negative, or Zero

Write a program that takes a number and prints whether it's positive, negative, or zero.

8. Login System

Create a simple login system that checks if a username and password match predetermined values. Use if-else to provide appropriate success or failure messages.

9. Ticket Price Calculator

Write a program for a movie theater that calculates ticket prices based on age:

- Children (0-12): $5
- Teenagers (13-17): $8
- Adults (18-64): $12
- Seniors (65+): $7

10. Day of the Week

Create a program that asks the user for a number between 1 and 7 and outputs the corresponding day of the week (1 is Monday, 7 is Sunday).

Intermediate Level (11-25)

11. BMI Calculator with Categories

Create a BMI (Body Mass Index) calculator that takes height (in meters) and weight (in kilograms) and categorizes the result as:

- Underweight: <18.5
- Normal weight: 18.5-24.9
- Overweight: 25-29.9
- Obesity: ≥30

12. Rock, Paper, Scissors

Implement a rock, paper, scissors game where the user plays against the computer. Use the ternary operator to determine the winner concisely.

13. Triangle Type Classifier

Write a program that takes the lengths of three sides of a triangle and determines if it's equilateral (all sides equal), isosceles (two sides equal), or scalene (no sides equal).

14. Tax Calculator

Create a program that calculates income tax based on the following brackets:

- $0-$10,000: 10%
- $10,001-$50,000: 20%

- $50,001-$100,000: 30%
- $100,001+: 40%

15. Password Strength Checker

Write a program that evaluates password strength based on:

- Length (at least 8 characters)
- Contains uppercase letters
- Contains lowercase letters
- Contains numbers
- Contains special characters Use nested if statements to check each condition and provide a rating (weak, medium, strong).

16. Season Determiner

Create a program that asks for a month and day and outputs the corresponding season. Use nested if statements or elif chains.

17. Coffee Shop Order System

Implement a coffee shop order system with nested conditions:

1. First, ask for the base drink (coffee, tea, or chocolate)
2. Based on the first selection, offer specific options (e.g., for coffee: americano, latte, espresso)
3. Ask if they want extras (milk, sugar, flavor shots)
4. Calculate and display the final price

18. Color Mixer

Write a program that asks the user for two primary colors (red, blue, yellow) and tells them what secondary color they would create when mixed. Use a nested structure.

19. Number Guessing Game with Hints

Create a number guessing game where the program picks a random number between 1 and 100. Give hints (higher/lower) using the ternary operator.

20. Shipping Cost Calculator

Implement a shipping calculator that determines cost based on:

- Package weight
- Destination (domestic/international)

- Shipping speed (standard/express/overnight) Use nested conditions to calculate the final cost.

21. Date Validator

Create a program that validates if a given date (day, month, year) is valid, considering leap years and the varying number of days in each month.

22. Flight Booking System

Implement a simple flight booking system that:

- Checks seat availability
- Applies discounts based on age or membership status
- Calculates final ticket price Use nested if statements and the ternary operator where appropriate.

23. Water State Determiner

Write a program that takes temperature (in Celsius) and atmospheric pressure and determines the state of water (solid, liquid, gas) at those conditions. Use nested if statements.

24. Credit Card Validator

Create a program that validates a credit card number based on:

- Length
- Starting digits (e.g., 4 for Visa, 5 for MasterCard)
- Luhn algorithm check Use if-elif-else chains to validate each condition.

25. Match-Case Menu System (Python 3.10+)

Implement a simple menu system using match-case that displays different options and performs different actions based on user selection.

Advanced Level (26-40)

26. Employee Payroll System

Create a program that calculates an employee's paycheck based on:

- Hours worked (regular and overtime)
- Position (determines hourly rate)
- Performance rating (determines bonus)

- Tax withholding based on income brackets Use nested conditions and the ternary operator for efficiency.

27. Loan Approval System

Implement a loan approval system that decides whether to approve a loan based on:

- Credit score
- Income
- Requested loan amount
- Current debt
- Employment history Use nested conditions with multiple factors.

28. Weather Advisory System

Create a program that issues weather advisories based on:

- Temperature (high/low)
- Precipitation type and amount
- Wind speed
- Visibility Use nested conditions to determine the type and severity of advisories.

29. Restaurant Table Reservation

Implement a restaurant reservation system that uses match-case (Python 3.10+) to handle different reservation scenarios:

- New reservation
- Modify existing reservation
- Cancel reservation
- Check availability For each case, implement specific logic with nested conditions.

30. Smart Home Control System

Create a smart home control system simulator that uses if-elif-else chains and match-case to:

- Adjust lighting based on time of day and occupancy
- Control temperature based on weather and preferences
- Manage security systems based on occupancy status
- Handle voice commands with pattern matching