50 Python Programming Tasks with Hints

Section 1: If-Else Statements (Tasks 1-12)

Task 1: Number Sign Checker

Problem: Write a program that asks the user for a number and tells them if it's positive, negative, or zero. **Hint:** Use if, elif, and else statements. Remember that 0 is neither positive nor negative.

Task 2: Age Category

Problem: Create a program that categorizes a person's age into: Child (0-12), Teen (13-19), Adult (20-64), Senior (65+). **Hint:** Use multiple elif statements to check age ranges.

Task 3: Grade Calculator

Problem: Input a numerical score (0-100) and output the letter grade (A: 90-100, B: 80-89, C: 70-79, D: 60-69, F: below 60). **Hint:** Start checking from the highest grade downward using if and elif.

Task 4: Even or Odd

Problem: Write a program that determines if a given number is even or odd. **Hint:** Use the modulo operator %. If number % 2 == 0, it's even.

Task 5: Leap Year Checker

Problem: Determine if a given year is a leap year. (Divisible by 4, but not by 100, unless also divisible by 400) **Hint:** Use logical operators and, or, and not to combine conditions.

Task 6: Temperature Advisor

Problem: Ask for temperature and give clothing advice: Hot (>80°F), Warm (60-80°F), Cool (40-59°F), Cold (<40°F). **Hint:** Use comparison operators and elif statements for temperature ranges.

Task 7: Login System

Problem: Create a simple login that checks username and password. Give 3 attempts before locking out. **Hint:** Use a counter variable and a while loop with if-else inside.

Task 8: Triangle Type Classifier

Problem: Input three sides of a triangle and determine if it's equilateral, isosceles, or scalene. **Hint:** Compare the three sides using == operator in different combinations.

Task 9: BMI Calculator with Categories

Problem: Calculate BMI and categorize: Underweight (<18.5), Normal (18.5-24.9), Overweight (25-29.9), Obese (≥30). **Hint:** BMI = weight(kg) / height(m)². Use elif for different ranges.

Task 10: Voting Eligibility

Problem: Check if a person can vote based on age (≥18) and citizenship status. **Hint:** Use logical and operator to check both conditions.

Task 11: Password Strength Checker

Problem: Evaluate password strength: Weak (length < 6), Medium (6-8 chars), Strong (>8 chars with numbers). **Hint:** Use len() function and any(char.isdigit() for char in password) to check for numbers.

Task 12: Season Identifier

Problem: Input a month number (1-12) and output the season. **Hint:** Group months: Winter (12,1,2), Spring (3,4,5), Summer (6,7,8), Fall (9,10,11).

Section 2: User Input with Float (Tasks 13-25)

Task 13: Simple Calculator

Problem: Create a calculator that performs basic operations (+, -, *, /) on two float numbers. **Hint:** Use float(input()) to get decimal numbers. Handle division by zero with if-else.

Task 14: Circle Area and Circumference

Problem: Input radius as float and calculate both area and circumference of a circle. **Hint:** Import math module for pi: import math. Area = $\pi \times r^2$, Circumference = $2 \times \pi \times r$.

Task 15: Currency Converter

Problem: Convert dollars to other currencies using float exchange rates. **Hint:** Store exchange rates as float variables. Multiply dollar amount by the rate.

Task 16: Tip Calculator

Problem: Calculate tip amount and total bill based on bill amount and tip percentage (both floats). **Hint:** Tip = bill_amount × (tip_percentage / 100). Total = bill_amount + tip.

Task 17: Compound Interest Calculator

Problem: Calculate compound interest given principal, rate, time, and compounding frequency (all floats). **Hint:** Formula: A = P(1 + r/n)^(nt). Use ** for exponentiation.

Task 18: Distance Between Points

Problem: Calculate distance between two points (x1,y1) and (x2,y2) using float coordinates. **Hint:** Distance = $\sqrt{[(x2-x1)^2 + (y2-y1)^2]}$. Use math.sqrt() function.

Task 19: Fuel Efficiency Calculator

Problem: Calculate miles per gallon given distance (float) and fuel used (float). **Hint:** MPG = distance / fuel_used. Handle zero fuel case.

Task 20: Temperature Converter

Problem: Convert between Celsius, Fahrenheit, and Kelvin using float inputs. **Hint:** F = C×9/5 + 32, K = C + 273.15. Create a menu system.

Task 21: Loan Payment Calculator

Problem: Calculate monthly payment for a loan using principal, annual rate, and years (floats). **Hint:** Monthly rate = annual_rate/12/100. Use loan payment formula.

Task 22: Pizza Cost Per Person

Problem: Calculate cost per person when ordering pizzas, including tax and tip (all floats). **Hint:** Total cost = (pizza_cost + tax + tip) / number_of_people.

Task 23: GPA Calculator

Problem: Calculate GPA from course credits and grades (floats). Input multiple courses. **Hint:** GPA = total_grade_points / total_credits. Use a loop to input multiple courses.

Task 24: Body Fat Percentage

Problem: Calculate body fat percentage using weight, height, age (floats) and gender. **Hint:** Use different formulas for men and women. Research Navy method formula.

Task 25: Investment Growth Calculator

Problem: Calculate how long it takes for an investment to double given annual return rate (float). **Hint:** Use Rule of 72: Years ≈ 72 / annual_return_rate.

Section 3: While Loops (Tasks 26-38)

Task 26: Number Guessing Game

Problem: Create a guessing game where user tries to guess a random number. Continue until correct. **Hint:** Use import random and random.randint(). Use while loop with boolean condition.

Task 27: Sum Until Zero

Problem: Keep asking for numbers and sum them until user enters 0. **Hint:** Initialize sum = 0 before loop. Use while True: and break when input is 0.

Task 28: Countdown Timer

Problem: Ask for a starting number and count down to zero, displaying each number. **Hint:** Use while number > 0: and decrement the number each iteration.

Task 29: Password Validation Loop

Problem: Keep asking for password until user enters the correct one. **Hint:** Use while password != correct_password: and update password inside loop.

Task 30: Menu System

Problem: Create a menu that keeps showing options until user chooses to exit. **Hint:** Use while True: and break when user selects exit option.

Task 31: Factorial Calculator

Problem: Calculate factorial of a number using while loop. **Hint:** Initialize result = 1, use while loop to multiply from 1 to n.

Task 32: Fibonacci Sequence

Problem: Generate Fibonacci sequence up to n terms using while loop. **Hint:** Start with a=0, b=1. In loop: next = a+b, then update a=b, b=next.

Task 33: Digital Root Calculator

Problem: Keep summing digits of a number until single digit remains. **Hint:** Use while loop with condition while number > 9:. Sum digits using modulo and division.

Task 34: Bank Account Simulator

Problem: Simulate bank account with deposit/withdraw options. Continue until user quits. **Hint:** Use while loop for menu, track balance, validate transactions.

Task 35: Prime Number Checker

Problem: Check if a number is prime using while loop for divisibility testing. **Hint:** Check divisibility from 2 to √number using while loop.

Task 36: Collatz Conjecture

Problem: Apply Collatz rules (n/2 if even, 3n+1 if odd) until reaching 1. **Hint:** Use while n != 1: and apply rules based on n%2.

Task 37: Shopping Cart Total

Problem: Keep adding item prices until user enters 0, then show total with tax. **Hint:** Use while loop to input prices, accumulate total, calculate tax outside loop.

Task 38: Reverse Number

Problem: Reverse the digits of a number using while loop. **Hint:** Use modulo to get last digit, build reversed number, use integer division.

Section 4: For Loops (Tasks 39-50)

Task 39: Multiplication Table

Problem: Print multiplication table for a given number from 1 to 10. **Hint:** Use for i in range(1, 11): and print number * i.

Task 40: Sum of List

Problem: Calculate sum of numbers in a list using for loop. **Hint:** Initialize total = 0, use for num in list: and add each number.

Task 41: Character Counter

Problem: Count occurrences of each character in a string using for loop. **Hint:** Use dictionary to store counts. Loop through string with for char in string:.

Task 42: Pattern Printer

Problem: Print pyramid patterns using nested for loops. **Hint:** Outer loop for rows, inner loops for spaces and stars. Use print("*", end="").

Task 43: Prime Numbers in Range

Problem: Find all prime numbers between 1 and n using for loops. **Hint:** Outer loop for each number, inner loop to check divisibility.

Task 44: Grade Statistics

Problem: Input multiple grades and calculate average, highest, and lowest using for loop. **Hint:** Use for i in range(number_of_grades): to input grades into a list.

Task 45: Word Reverser

Problem: Reverse each word in a sentence while keeping word order. **Hint:** Split sentence into words, use for loop to reverse each word, join back.

Task 46: Number Pattern Generator

Problem: Generate number patterns (like 1,22,333,4444) using nested for loops. **Hint:** Outer loop for rows, inner loop to print digit repeated row-number times.

Task 47: List Comprehension vs For Loop

Problem: Create the same list using both for loop and list comprehension (squares of 1-10). **Hint:** For loop: append to empty list. List comp: [x**2 for x in range(1,11)].

Task 48: String Analyzer

Problem: Analyze a string: count vowels, consonants, digits, and spaces using for loop. **Hint:** Use counters and char.isalpha(), char.isdigit(), char.isspace() methods.

Task 49: Nested List Processor

Problem: Process a 2D list (list of lists) to find sum of each sublist. **Hint:** Outer loop for each sublist, inner loop to sum elements in current sublist.

Task 50: Final Challenge - Student Management System

Problem: Create a system that manages student records with grades. Use all concepts: if-else for validation, float input for grades, while loop for menu, for loop to process student lists. **Hint:** Combine all learned concepts. Create functions for different operations. Use dictionaries to store student data.