

Control Flow

Assignment Questions



Basic If-Else Statements:

1. Write a Python program to check if a given number is positive or negative.
2. Create a program that determines if a person is eligible to vote based on their age.
3. Develop a program to find the maximum of two numbers using if-else statements.
4. Write a Python script to classify a given year as a leap year or not.
5. Create a program that checks whether a character is a vowel or a consonant.
6. Implement a program to determine whether a given number is even or odd.
7. Write a Python function to calculate the absolute value of a number without using the `abs()` function.
8. Develop a program that determines the largest of three given numbers using if-else statements.
9. Create a program that checks if a given string is a palindrome.
10. Write a Python program to calculate the grade based on a student's score.

Nested If-Else Statements:

11. Write a program to find the largest among three numbers using nested if-else statements.
12. Implement a program to determine if a triangle is equilateral, isosceles, or scalene.
13. Develop a program that checks if a year is a leap year and also if it is a century year.
14. Write a Python script to determine if a number is positive, negative, or zero.
15. Create a program to check if a person is a teenager (between 13 and 19 years old).
16. Develop a program that determines the type of angle based on its measure (acute, obtuse, or right).
17. Write a Python program to calculate the roots of a quadratic equation.
18. Implement a program to determine the day of the week based on a user-provided number (1 for Monday, 2 for Tuesday, etc.).
19. Create a program that determines if a year is a leap year and also if it is evenly divisible by 400.
20. Develop a program that checks if a given number is prime or not using nested if-else statements.

Elif Statements:

21. Write a Python program to assign grades based on different ranges of scores using elif statements.
22. Implement a program to determine the type of a triangle based on its angles.
23. Develop a program to categorize a given person's BMI into underweight, normal, overweight, or obese using elif statements.
24. Create a program that determines whether a given number is positive, negative, or zero using elif statements.
25. Write a Python script to determine the type of a character (uppercase, lowercase, or special) using elif statements.
26. Implement a program to calculate the discounted price based on different purchase amounts using elif statements.
27. Develop a program to calculate the electricity bill based on different consumption slabs using elif statements.
28. Create a program to determine the type of quadrilateral based on its angles and sides using elif statements.
29. Write a Python script to determine the season based on a user-provided month using elif statements.
30. Implement a program to determine the type of a year (leap or common) and month (30 or 31 days) using elif statements.

Assignment Questions

Basic Level:

1. Write a Python program that checks if a given number is positive, negative, or zero.
2. Create a program to determine if a person is eligible to vote based on their age.
3. Write a program to find the maximum of two given numbers using conditional statements.
4. Develop a program that calculates the grade of a student based on their exam score.
5. Create a program that checks if a year is a leap year or not.
6. Write a program to classify a triangle based on its sides' lengths.
7. Build a program that determines the largest of three given numbers.
8. Develop a program that checks whether a character is a vowel or a consonant.
9. Create a program to calculate the total cost of a shopping cart based on discounts.
10. Write a program that checks if a given number is even or odd.

Intermediate Level:

11. Write a program that calculates the roots of a quadratic equation .
12. Create a program that determines the day of the week based on the day number (1-7).
13. Develop a program that calculates the factorial of a given number using recursion.
14. Write a program to find the largest among three numbers without using the `max()` function.
15. Create a program that simulates a basic ATM transaction menu.
16. Build a program that checks if a given string is a palindrome or not.
17. Write a program that calculates the average of a list of numbers, excluding the smallest and largest values.
18. Develop a program that converts a given temperature from Celsius to Fahrenheit.
19. Create a program that simulates a basic calculator for addition, subtraction, multiplication, and division.
20. Write a program that determines the roots of a cubic equation using the Cardano formula.

Advanced Level:

21. Create a program that calculates the income tax based on the user's income and tax brackets.
22. Write a program that simulates a rock-paper-scissors game against the computer.
23. Develop a program that generates a random password based on user preferences (length, complexity).
24. Create a program that implements a simple text-based adventure game with branching scenarios.
25. Build a program that solves a linear equation for x, considering different cases.
26. Write a program that simulates a basic quiz game with multiple-choice questions and scoring.
27. Develop a program that determines whether a given year is a prime number or not.
28. Create a program that sorts three numbers in ascending order using conditional statements.
29. Build a program that determines the roots of a quartic equation using numerical methods.
30. Write a program that calculates the BMI (Body Mass Index) and provides health recommendations based on the user's input.

Challenge Level:

31. Create a program that validates a password based on complexity rules (length, characters, etc.).
32. Develop a program that performs matrix addition and subtraction based on user input.
33. Write a program that calculates the greatest common divisor (GCD) of two numbers using the Euclidean algorithm.
34. Build a program that performs matrix multiplication using nested loops and conditional statements.
35. Create a program that simulates a basic text-based tic-tac-toe game against the computer.
36. Write a program that generates Fibonacci numbers up to a specified term using iterative methods.
37. Develop a program that calculates the nth term of the Fibonacci sequence using memoization.
38. Create a program that generates a calendar for a given month and year using conditional statements.
39. Build a program that simulates a basic text-based blackjack game against the computer.
40. Write a program that generates the prime factors of a given number using trial division.