100 Python Problems for Beginners

Complete Programming Practice Guide

Table of Contents

Section 1: Basic Syntax & Variables (Problems 1-10)

Problem 1: Write a program to print "Hello, World!" to the console.

Problem 2: Create variables to store your name, age, and favorite color, then print them.

Problem 3: Write a program that takes two numbers as input and prints their sum.

Problem 4: Create a program that calculates the area of a rectangle given length and width.

Problem 5: Write a program to swap the values of two variables without using a third variable.

Problem 6: Create a program that converts temperature from Celsius to Fahrenheit.

Problem 7: Write a program to calculate simple interest given principal, rate, and time.

Problem 8: Create a program that takes a number as input and prints its square and cube.

Problem 9: Write a program to calculate the perimeter of a circle given its radius.

Problem 10: Create a program that takes three numbers and finds their average.

Section 2: Data Types & Operations (Problems 11-20)

Problem 11: Write a program to check the data type of different variables.

Problem 12: Create a program that performs all arithmetic operations on two numbers.

Problem 13: Write a program to check if a number is even or odd using the modulus operator.

Problem 14: Create a program that converts minutes to hours and minutes.

Problem 15: Write a program to calculate compound interest.

Problem 16: Create a program that finds the remainder when one number is divided by another.

Problem 17: Write a program to check if a year is a leap year.

Problem 18: Create a program that calculates the distance between two points (x1,y1) and (x2,y2).

Problem 19: Write a program to convert seconds into hours, minutes, and seconds.

Problem 20: Create a program that checks if a number is positive, negative, or zero.

Section 3: Conditional Statements (Problems 21-30)

Problem 21: Write a program to find the largest of three numbers.

Problem 22: Create a program that determines if a person is eligible to vote (age >= 18).

Problem 23: Write a program to check if a triangle is valid given three sides.

Problem 24: Create a program that assigns letter grades based on numerical scores.

Problem 25: Write a program to determine if a character is a vowel or consonant.

Problem 26: Create a program that checks if a number is divisible by both 3 and 5.

Problem 27: Write a program to find the maximum of four numbers.

Problem 28: Create a program that determines the type of triangle (equilateral, isosceles, scalene).

Problem 29: Write a program to check if a person qualifies for a loan based on age and income.

Problem 30: Create a program that determines the quadrant of a point in a coordinate system.

Section 4: Loops (Problems 31-45)

Problem 31: Write a program to print numbers from 1 to 10 using a for loop.

Problem 32: Create a program to print the multiplication table of a given number.

Problem 33: Write a program to find the sum of first n natural numbers.

Problem 34: Create a program to print all even numbers between 1 and 50.

Problem 35: Write a program to find the factorial of a number using a loop.

Problem 36: Create a program to count the number of digits in a number.

Problem 37: Write a program to reverse a number.

Problem 38: Create a program to check if a number is prime.

Problem 39: Write a program to find the sum of digits of a number.

Problem 40: Create a program to print the Fibonacci series up to n terms.

Problem 41: Write a program to find all prime numbers between 1 and 100.

Problem 42: Create a program to find the GCD of two numbers.

Problem 43: Write a program to print a pattern of stars forming a triangle.

Problem 44: Create a program to find the LCM of two numbers.

Problem 45: Write a program to check if a number is an Armstrong number.

Section 5: Functions (Problems 46-55)

Problem 46: Write a function to calculate the area of a circle.

Problem 47: Create a function that checks if a number is prime.

Problem 48: Write a function to find the maximum of three numbers.

Problem 49: Create a function that converts temperature between Celsius and Fahrenheit.

Problem 50: Write a function to calculate the power of a number (x^n).

Problem 51: Create a function that checks if a string is a palindrome.

Problem 52: Write a function to generate the Fibonacci sequence.

Problem 53: Create a function that counts vowels in a string.

Problem 54: Write a function to find the factorial of a number using recursion.

Problem 55: Create a function that returns both quotient and remainder of division.

Section 6: Lists & Tuples (Problems 56-70)

Problem 56: Write a program to find the sum of all elements in a list.

Problem 57: Create a program to find the largest element in a list.

Problem 58: Write a program to count the occurrences of an element in a list.

Problem 59: Create a program to reverse a list.

Problem 60: Write a program to remove duplicates from a list.

Problem 61: Create a program to find the second largest number in a list.

Problem 62: Write a program to merge two lists.

Problem 63: Create a program to find common elements between two lists.

Problem 64: Write a program to sort a list in ascending order without using sort().

Problem 65: Create a program to find the index of an element in a list.

Problem 66: Write a program to create a list of squares of numbers from 1 to 10.

Problem 67: Create a program to flatten a nested list.

Problem 68: Write a program to find pairs of numbers that sum to a target value.

Problem 69: Create a program to rotate a list by k positions.

Problem 70: Write a program to find the intersection of two lists.

Section 7: Dictionaries & Sets (Problems 71-80)

Problem 71: Write a program to count the frequency of words in a sentence.

Problem 72: Create a program to merge two dictionaries.

Problem 73: Write a program to find the key with the maximum value in a dictionary.

Problem 74: Create a program to invert a dictionary (swap keys and values).

Problem 75: Write a program to group students by their grades using dictionaries.

Problem 76: Create a program to find common keys between two dictionaries.

Problem 77: Write a program to remove duplicates from a list using sets.

Problem 78: Create a program to find the union and intersection of two sets.

Problem 79: Write a program to check if one set is a subset of another.

Problem 80: Create a program to find elements that are in one set but not in another.

Section 8: Strings (Problems 81-90)

Problem 81: Write a program to count the number of words in a string.

Problem 82: Create a program to reverse each word in a sentence.

Problem 83: Write a program to check if two strings are anagrams.

Problem 84: Create a program to find the longest word in a sentence.

Problem 85: Write a program to capitalize the first letter of each word.

Problem 86: Create a program to remove all vowels from a string.

Problem 87: Write a program to count the occurrences of each character in a string.

Problem 88: Create a program to replace all spaces with underscores.

Problem 89: Write a program to check if a string contains only digits.

Problem 90: Create a program to find the most frequent character in a string.

Section 9: File Handling & Modules (Problems 91-95)

Problem 91: Write a program to read a text file and count the number of lines.

Problem 92: Create a program to write a list of names to a file.

Problem 93: Write a program to copy contents from one file to another.

Problem 94: Create a program to find and replace text in a file.

Problem 95: Write a program using the random module to simulate dice rolls.

Section 10: Mixed Challenges (Problems 96-100)

Problem 96: Create a simple calculator that can perform basic operations.

Problem 97: Write a program to create a simple guessing game.

Problem 98: Create a program to manage a simple phone book (add, search, delete contacts).

Problem 99: Write a program to find the longest common subsequence of two strings.

Problem 100: Create a program that implements a simple banking system with deposit, withdraw, and balance check operations.

Solutions Guide

Sample Solutions for First Few Problems:

Solution 1:

python

```python
print("Hello, World!")
```

Solution 2:

python

```python
name = "Alice"
age = 25
favorite_color = "blue"
print(f"Name: {name}, Age: {age}, Favorite Color: {favorite_color}")
```

Solution 3:

python

```python
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
sum_result = num1 + num2
print(f"The sum is: {sum_result}")
```

Solution 4:

```python
length = float(input("Enter length: "))
width = float(input("Enter width: "))
area = length * width
print(f"Area of rectangle: {area}")
```

Solution 5:

```python
a = 10
b = 20
print(f"Before swap: a = {a}, b = {b}")
a, b = b, a
print(f"After swap: a = {a}, b = {b}")
```

Practice Tips for Beginners

1. **Start Simple**: Begin with basic problems and gradually increase complexity.
2. **Practice Daily**: Solve at least 2-3 problems every day to build consistency.
3. **Understand Before Moving On**: Make sure you understand each solution before proceeding.
4. **Write Clean Code**: Focus on writing readable and well-commented code.
5. **Test Your Solutions**: Always test your programs with different inputs.
6. **Learn from Mistakes**: Debug errors carefully to understand what went wrong.
7. **Use Resources**: Don't hesitate to look up documentation when needed.
8. **Code Reviews**: Try to optimize your solutions and find alternative approaches.

Additional Resources

- **Python Documentation**: https://docs.python.org/
- **Online Python Interpreter**: https://repl.it/
- **Python Style Guide**: PEP 8
- **Practice Platforms**: LeetCode, HackerRank, Codewars

Conclusion

These 100 problems cover all fundamental concepts in Python programming. Work through them systematically, and you'll build a strong foundation in Python. Remember, the key to becoming proficient in programming is consistent practice and patience.

Good luck with your Python journey!
