

100 Beginner-Friendly Python Practice Questions

Basic Output & Variables (Questions 1-10)

1. Hello World

Write a function that returns "Hello, World!"

```
python

def hello_world():
    # Your code here
    pass
```

Test Cases:

- Input: `hello_world()` → Output: `"Hello, World!"`

2. Personal Greeting

Write a function that takes a name and returns "Hello, [name]!"

```
python

def greet(name):
    # Your code here
    pass
```

Test Cases:

- Input: `greet("Alice")` → Output: `"Hello, Alice!"`
- Input: `greet("Bob")` → Output: `"Hello, Bob!"`

3. Add Two Numbers

Write a function that adds two numbers and returns the result.

```
python

def add(a, b):
    # Your code here
    pass
```

Test Cases:

- Input: `add(5, 3)` → Output: `8`
- Input: `add(-1, 1)` → Output: `0`
- Input: `add(0, 0)` → Output: `0`

4. Square a Number

Write a function that returns the square of a number.

python

```
def square(n):
    # Your code here
    pass
```

Test Cases:

- Input: `square(5)` → Output: `25`
- Input: `square(-3)` → Output: `9`
- Input: `square(0)` → Output: `0`

5. Calculate Area of Rectangle

Write a function that calculates the area of a rectangle given length and width.

python

```
def rectangle_area(length, width):
    # Your code here
    pass
```

Test Cases:

- Input: `rectangle_area(5, 3)` → Output: `15`
- Input: `rectangle_area(10, 2)` → Output: `20`

6. Convert Celsius to Fahrenheit

Write a function that converts Celsius to Fahrenheit using the formula: $F = (C \times 9/5) + 32$

python

```
def celsius_to_fahrenheit(celsius):
    # Your code here
    pass
```

Test Cases:

- Input: `celsius_to_fahrenheit(0)` → Output: `32.0`
- Input: `celsius_to_fahrenheit(100)` → Output: `212.0`
- Input: `celsius_to_fahrenheit(-40)` → Output: `-40.0`

7. Calculate Average

Write a function that returns the average of three numbers.

```
python

def average(a, b, c):
    # Your code here
    pass
```

Test Cases:

- Input: `average(10, 20, 30)` → Output: `20.0`
- Input: `average(5, 5, 5)` → Output: `5.0`

8. Swap Two Variables

Write a function that swaps two values and returns them as a tuple.

```
python

def swap(a, b):
    # Your code here
    pass
```

Test Cases:

- Input: `swap(1, 2)` → Output: `(2, 1)`
- Input: `swap("hello", "world")` → Output: `("world", "hello")`

9. Calculate Circle Area

Write a function that calculates the area of a circle given its radius. Use $\pi = 3.14159$

```
python
```

```
def circle_area(radius):  
    # Your code here  
    pass
```

Test Cases:

- Input: `circle_area(1)` → Output: `3.14159` (approximately)
- Input: `circle_area(5)` → Output: `78.53975` (approximately)

10. Minutes to Seconds

Write a function that converts minutes to seconds.

```
python
```

```
def minutes_to_seconds(minutes):  
    # Your code here  
    pass
```

Test Cases:

- Input: `minutes_to_seconds(5)` → Output: `300`
- Input: `minutes_to_seconds(1)` → Output: `60`

Conditionals (Questions 11-25)

11. Check Even or Odd

Write a function that returns "Even" if a number is even, "Odd" if odd.

```
python
```

```
def even_or_odd(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `even_or_odd(4)` → Output: `"Even"`
- Input: `even_or_odd(7)` → Output: `"Odd"`
- Input: `even_or_odd(0)` → Output: `"Even"`

12. Find Maximum of Two Numbers

Write a function that returns the larger of two numbers.

```
python
```

```
def max_of_two(a, b):  
    # Your code here  
    pass
```

Test Cases:

- Input: `max_of_two(5, 10)` → Output: `10`
- Input: `max_of_two(-1, -5)` → Output: `-1`
- Input: `max_of_two(3, 3)` → Output: `3`

13. Check Positive, Negative, or Zero

Write a function that returns "Positive", "Negative", or "Zero".

```
python
```

```
def check_sign(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `check_sign(5)` → Output: `"Positive"`
- Input: `check_sign(-3)` → Output: `"Negative"`
- Input: `check_sign(0)` → Output: `"Zero"`

14. Grade Calculator

Write a function that returns a grade based on score: A (90+), B (80-89), C (70-79), D (60-69), F (<60)

```
python
```

```
def calculate_grade(score):  
    # Your code here  
    pass
```

Test Cases:

- Input: `calculate_grade(95)` → Output: `"A"`
- Input: `calculate_grade(82)` → Output: `"B"`
- Input: `calculate_grade(55)` → Output: `"F"`

15. Leap Year Checker

Write a function that returns True if a year is a leap year, False otherwise.

python

```
def is_leap_year(year):
    # Your code here
    pass
```

Test Cases:

- Input: `is_leap_year(2020)` → Output: `True`
- Input: `is_leap_year(2021)` → Output: `False`
- Input: `is_leap_year(2000)` → Output: `True`
- Input: `is_leap_year(1900)` → Output: `False`

16. Check Vowel or Consonant

Write a function that returns "Vowel" if the character is a vowel, "Consonant" otherwise.

python

```
def vowel_or_consonant(char):
    # Your code here
    pass
```

Test Cases:

- Input: `vowel_or_consonant('a')` → Output: `"Vowel"`
- Input: `vowel_or_consonant('b')` → Output: `"Consonant"`
- Input: `vowel_or_consonant('E')` → Output: `"Vowel"`

17. Absolute Value

Write a function that returns the absolute value of a number (without using `abs()`).

python

```
def absolute(n):
    # Your code here
    pass
```

Test Cases:

- Input: `absolute(-5)` → Output: `5`
- Input: `absolute(3)` → Output: `3`
- Input: `absolute(0)` → Output: `0`

18. Can Vote Checker

Write a function that returns True if age ≥ 18 , False otherwise.

```
python

def can_vote(age):
    # Your code here
    pass
```

Test Cases:

- Input: `can_vote(20)` → Output: `True`
- Input: `can_vote(17)` → Output: `False`
- Input: `can_vote(18)` → Output: `True`

19. Triangle Type Checker

Write a function that returns "Equilateral", "Isosceles", or "Scalene" based on three sides.

```
python

def triangle_type(a, b, c):
    # Your code here
    pass
```

Test Cases:

- Input: `triangle_type(3, 3, 3)` → Output: `"Equilateral"`
- Input: `triangle_type(3, 3, 5)` → Output: `"Isosceles"`
- Input: `triangle_type(3, 4, 5)` → Output: `"Scalene"`

20. Max of Three Numbers

Write a function that returns the maximum of three numbers.

```
python
```

```
def max_of_three(a, b, c):  
    # Your code here  
    pass
```

Test Cases:

- Input: `max_of_three(1, 2, 3)` → Output: `3`
- Input: `max_of_three(10, 5, 8)` → Output: `10`
- Input: `max_of_three(7, 7, 7)` → Output: `7`

21. Check Divisibility

Write a function that returns True if the first number is divisible by the second.

```
python
```

```
def is_divisible(n, divisor):  
    # Your code here  
    pass
```

Test Cases:

- Input: `is_divisible(10, 2)` → Output: `True`
- Input: `is_divisible(10, 3)` → Output: `False`
- Input: `is_divisible(0, 5)` → Output: `True`

22. BMI Calculator

Write a function that calculates BMI and returns "Underweight" (<18.5), "Normal" (18.5-24.9), or "Overweight" (≥ 25)

```
python
```

```
def bmi_category(weight, height):  
    # Your code here (BMI = weight / height^2)  
    pass
```

Test Cases:

- Input: `bmi_category(70, 1.75)` → Output: `"Normal"`
- Input: `bmi_category(50, 1.75)` → Output: `"Underweight"`

23. Check Range

Write a function that returns True if a number is between 1 and 100 (inclusive).

```
python
```

```
def in_range(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `in_range(50)` → Output: `True`
- Input: `in_range(0)` → Output: `False`
- Input: `in_range(100)` → Output: `True`

24. Password Strength

Write a function that returns "Weak" if password length < 6, "Strong" otherwise.

```
python
```

```
def password_strength(password):  
    # Your code here  
    pass
```

Test Cases:

- Input: `password_strength("abc")` → Output: `"Weak"`
- Input: `password_strength("secure123")` → Output: `"Strong"`

25. Season Checker

Write a function that returns the season based on month (12-2: Winter, 3-5: Spring, 6-8: Summer, 9-11: Fall)

```
python
```

```
def get_season(month):  
    # Your code here  
    pass
```

Test Cases:

- Input: `get_season(1)` → Output: `"Winter"`
- Input: `get_season(6)` → Output: `"Summer"`
- Input: `get_season(10)` → Output: `"Fall"`

Loops (Questions 26-45)

26. Sum of First N Numbers

Write a function that returns the sum of numbers from 1 to n.

```
python
```

```
def sum_n(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `sum_n(5)` → Output: `15` ($1+2+3+4+5$)
- Input: `sum_n(10)` → Output: `55`
- Input: `sum_n(1)` → Output: `1`

27. Factorial

Write a function that returns the factorial of n.

```
python
```

```
def factorial(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `factorial(5)` → Output: `120`
- Input: `factorial(0)` → Output: `1`
- Input: `factorial(3)` → Output: `6`

28. Multiplication Table

Write a function that returns a list of the first 10 multiples of n.

```
python
```

```
def multiplication_table(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `multiplication_table(5)` → Output: `[5, 10, 15, 20, 25, 30, 35, 40, 45, 50]`

29. Count Digits

Write a function that counts the number of digits in a number.

```
python
```

```
def count_digits(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `count_digits(12345)` → Output: `5`
- Input: `count_digits(7)` → Output: `1`
- Input: `count_digits(0)` → Output: `1`

30. Reverse Number

Write a function that reverses the digits of a number.

```
python
```

```
def reverse_number(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `reverse_number(12345)` → Output: `54321`
- Input: `reverse_number(100)` → Output: `1`

31. Check Prime Number

Write a function that returns True if a number is prime.

```
python
```

```
def is_prime(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `is_prime(7)` → Output: `True`
- Input: `is_prime(10)` → Output: `False`
- Input: `is_prime(2)` → Output: `True`
- Input: `is_prime(1)` → Output: `False`

32. Fibonacci Sequence

Write a function that returns the first n Fibonacci numbers.

```
python
```

```
def fibonacci(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `fibonacci(5)` → Output: `[0, 1, 1, 2, 3]`
- Input: `fibonacci(7)` → Output: `[0, 1, 1, 2, 3, 5, 8]`

33. Sum of Even Numbers

Write a function that returns the sum of even numbers from 1 to n.

```
python
```

```
def sum_even(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `sum_even(10)` → Output: `30` ($2+4+6+8+10$)
- Input: `sum_even(5)` → Output: `6` ($2+4$)

34. Power Function

Write a function that calculates base raised to exponent (without using `**`).

```
python
```

```
def power(base, exp):  
    # Your code here  
    pass
```

Test Cases:

- Input: `power(2, 3)` → Output: `8`
- Input: `power(5, 0)` → Output: `1`
- Input: `power(3, 4)` → Output: `81`

35. GCD (Greatest Common Divisor)

Write a function that finds the GCD of two numbers.

```
python
```

```
def gcd(a, b):  
    # Your code here  
    pass
```

Test Cases:

- Input: `gcd(48, 18)` → Output: `6`
- Input: `gcd(100, 50)` → Output: `50`

36. Print Pattern

Write a function that returns a list of strings forming a right triangle pattern of stars.

```
python
```

```
def star_pattern(n):  
    # Your code here  
    # Return list like ["*", "**", "***"] for n=3  
    pass
```

Test Cases:

- Input: `star_pattern(3)` → Output: `["*", "**", "***"]`

- Input: `star_pattern(5)` → Output: `[["*", "**", "***", "****", "*****"]]`

37. Sum of Digits

Write a function that returns the sum of digits of a number.

```
python
```

```
def sum_digits(n):
    # Your code here
    pass
```

Test Cases:

- Input: `sum_digits(123)` → Output: `6`
- Input: `sum_digits(9876)` → Output: `30`

38. Armstrong Number

Write a function that checks if a number is an Armstrong number (sum of cubes of digits equals the number).

```
python
```

```
def is_armstrong(n):
    # Your code here
    pass
```

Test Cases:

- Input: `is_armstrong(153)` → Output: `True` ($1^3 + 5^3 + 3^3 = 153$)
- Input: `is_armstrong(370)` → Output: `True`
- Input: `is_armstrong(100)` → Output: `False`

39. LCM (Least Common Multiple)

Write a function that finds the LCM of two numbers.

```
python
```

```
def lcm(a, b):
    # Your code here
    pass
```

Test Cases:

- Input: `lcm(12, 18)` → Output: `36`
- Input: `lcm(5, 7)` → Output: `35`

40. Count Vowels in Range

Write a function that counts vowels in a string.

```
python
```

```
def count_vowels(s):  
    # Your code here  
    pass
```

Test Cases:

- Input: `count_vowels("hello")` → Output: `2`
- Input: `count_vowels("aeiou")` → Output: `5`

41. Perfect Number

Write a function that checks if a number is perfect (sum of divisors equals the number).

```
python
```

```
def is_perfect(n):  
    # Your code here  
    pass
```

Test Cases:

- Input: `is_perfect(6)` → Output: `True` ($1+2+3=6$)
- Input: `is_perfect(28)` → Output: `True`
- Input: `is_perfect(12)` → Output: `False`

42. Binary to Decimal

Write a function that converts a binary string to decimal.

```
python
```

```
def binary_to_decimal(binary):  
    # Your code here  
    pass
```

Test Cases:

- Input: `binary_to_decimal("1010")` → Output: `10`
- Input: `binary_to_decimal("1111")` → Output: `15`

43. Decimal to Binary

Write a function that converts a decimal number to binary (as a string).

```
python
```

```
def decimal_to_binary(n):
    # Your code here
    pass
```

Test Cases:

- Input: `decimal_to_binary(10)` → Output: `"1010"`
- Input: `decimal_to_binary(15)` → Output: `"1111"`

44. Palindrome Number

Write a function that checks if a number is a palindrome.

```
python
```

```
def is_palindrome_number(n):
    # Your code here
    pass
```

Test Cases:

- Input: `is_palindrome_number(121)` → Output: `True`
- Input: `is_palindrome_number(123)` → Output: `False`

45. Count Occurrences

Write a function that counts how many times a digit appears in a number.

```
python
```

```
def count_digit(n, digit):
    # Your code here
    pass
```

Test Cases:

- Input: `count_digit(112233, 2)` → Output: `2`
- Input: `count_digit(555, 5)` → Output: `3`

Strings (Questions 46-65)

46. String Length

Write a function that returns the length of a string (without using `len()`).

```
python

def string_length(s):
    # Your code here
    pass
```

Test Cases:

- Input: `string_length("hello")` → Output: `5`
- Input: `string_length("")` → Output: `0`

47. Reverse String

Write a function that reverses a string.

```
python

def reverse_string(s):
    # Your code here
    pass
```

Test Cases:

- Input: `reverse_string("hello")` → Output: `"olleh"`
- Input: `reverse_string("python")` → Output: `"nohtyp"`

48. Check Palindrome String

Write a function that checks if a string is a palindrome.

```
python
```

```
def is_palindrome(s):
    # Your code here
    pass
```

Test Cases:

- Input: `is_palindrome("racecar")` → Output: `True`
- Input: `is_palindrome("hello")` → Output: `False`
- Input: `is_palindrome("madam")` → Output: `True`

49. Count Words

Write a function that counts the number of words in a string.

```
python

def count_words(s):
    # Your code here
    pass
```

Test Cases:

- Input: `count_words("Hello World")` → Output: `2`
- Input: `count_words("Python is fun")` → Output: `3`

50. Capitalize First Letter

Write a function that capitalizes the first letter of each word.

```
python

def capitalize_words(s):
    # Your code here
    pass
```

Test Cases:

- Input: `capitalize_words("hello world")` → Output: `"Hello World"`
- Input: `capitalize_words("python programming")` → Output: `"Python Programming"`

51. Remove Spaces

Write a function that removes all spaces from a string.

```
python
```

```
def remove_spaces(s):  
    # Your code here  
    pass
```

Test Cases:

- Input: `remove_spaces("hello world")` → Output: `"helloworld"`
- Input: `remove_spaces("a b c")` → Output: `"abc"`

52. Count Character

Write a function that counts occurrences of a character in a string.

```
python
```

```
def count_char(s, char):  
    # Your code here  
    pass
```

Test Cases:

- Input: `count_char("hello", "l")` → Output: `(2)`
- Input: `count_char("mississippi", "s")` → Output: `(4)`

53. First Character

Write a function that returns the first character of a string.

```
python
```

```
def first_char(s):  
    # Your code here  
    pass
```

Test Cases:

- Input: `first_char("hello")` → Output: `"h"`
- Input: `first_char("python")` → Output: `"p"`

54. Last Character

Write a function that returns the last character of a string.

```
python
```

```
def last_char(s):  
    # Your code here  
    pass
```

Test Cases:

- Input: `last_char("hello")` → Output: `"o"`
- Input: `last_char("python")` → Output: `"n"`

55. String Contains

Write a function that checks if a substring exists in a string.

```
python
```

```
def contains(s, substring):  
    # Your code here  
    pass
```

Test Cases:

- Input: `contains("hello world", "world")` → Output: `True`
- Input: `contains("hello world", "bye")` → Output: `False`

56. Replace Character

Write a function that replaces all occurrences of a character with another.

```
python
```

```
def replace_char(s, old, new):  
    # Your code here  
    pass
```

Test Cases:

- Input: `replace_char("hello", "l", "r")` → Output: `"herro"`
- Input: `replace_char("aaa", "a", "b")` → Output: `"bbb"`

57. Is Uppercase

Write a function that checks if all letters in a string are uppercase.

```
python
```

```
def is_all_uppercase(s):  
    # Your code here  
    pass
```

Test Cases:

- Input: `is_all_uppercase("HELLO")` → Output: `True`
- Input: `is_all_uppercase("Hello")` → Output: `False`

58. Is Lowercase

Write a function that checks if all letters in a string are lowercase.

```
python
```

```
def is_all_lowercase(s):  
    # Your code here  
    pass
```

Test Cases:

- Input: `is_all_lowercase("hello")` → Output: `True`
- Input: `is_all_lowercase("Hello")` → Output: `False`

59. Concatenate Strings

Write a function that joins a list of strings with a separator.

```
python
```

```
def join_strings(strings, separator):  
    # Your code here  
    pass
```

Test Cases:

- Input: `join_strings(["a", "b", "c"], "-")` → Output: `"a-b-c"`
- Input: `join_strings(["hello", "world"], " ")` → Output: `"hello world"`

60. Remove Character

Write a function that removes all occurrences of a character from a string.

```
python
```

```
def remove_char(s, char):  
    # Your code here  
    pass
```

Test Cases:

- Input: `remove_char("hello", "l")` → Output: `"heo"`
- Input: `remove_char("aabbcc", "b")` → Output: `"aacc"`

61. Anagram Check

Write a function that checks if two strings are anagrams.

```
python
```

```
def are_anagrams(s1, s2):  
    # Your code here  
    pass
```

Test Cases:

- Input: `are_anagrams("listen", "silent")` → Output: `True`
- Input: `are_anagrams("hello", "world")` → Output: `False`

62. Count Consonants

Write a function that counts consonants in a string.

```
python
```

```
def count_consonants(s):  
    # Your code here  
    pass
```

Test Cases:

- Input: `count_consonants("hello")` → Output: `3`
- Input: `count_consonants("aeiou")` → Output: `0`

63. Swap Case

Write a function that swaps the case of all letters (upper to lower, lower to upper).

```
python
```

```
def swap_case(s):  
    # Your code here  
    pass
```

Test Cases:

- Input: `swap_case("Hello")` → Output: `"hELLO"`
- Input: `swap_case("PyThOn")` → Output: `"pYtHoN"`

64. Starts With

Write a function that checks if a string starts with a given prefix.

```
python
```

```
def starts_with(s, prefix):  
    # Your code here  
    pass
```

Test Cases:

- Input: `starts_with("hello", "he")` → Output: `True`
- Input: `starts_with("hello", "hi")` → Output: `False`

65. Ends With

Write a function that checks if a string ends with a given suffix.

```
python
```

```
def ends_with(s, suffix):  
    # Your code here  
    pass
```

Test Cases:

- Input: `ends_with("hello", "lo")` → Output: `True`
- Input: `ends_with("hello", "he")` → Output: `False`

Lists (Questions 66-85)

66. Sum of List

Write a function that returns the sum of all elements in a list.

```
python

def sum_list(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `sum_list([1, 2, 3, 4])` → Output: `10`
- Input: `sum_list([10, 20])` → Output: `30`

67. Average of List

Write a function that returns the average of a list.

```
python

def average_list(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `average_list([1, 2, 3, 4])` → Output: `2.5`
- Input: `average_list([10, 20, 30])` → Output: `20.0`

68. Maximum in List

Write a function that finds the maximum value in a list (without using `max()`).

```
python

def find_max(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `find_max([1, 5, 3, 9, 2])` → Output: `9`
- Input: `find_max([-1, -5, -3])` → Output: `-1`

69. Minimum in List

Write a function that finds the minimum value in a list (without using min()).

```
python
```

```
def find_min(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `find_min([1, 5, 3, 9, 2])` → Output: `1`
- Input: `find_min([-1, -5, -3])` → Output: `-5`

70. Reverse List

Write a function that reverses a list (without using reverse()).

```
python
```

```
def reverse_list(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `reverse_list([1, 2, 3, 4])` → Output: `[4, 3, 2, 1]`
- Input: `reverse_list(['a', 'b', 'c'])` → Output: `['c', 'b', 'a']`

71. Count Element

Write a function that counts occurrences of an element in a list.

```
python
```

```
def count_element(lst, element):
    # Your code here
    pass
```

Test Cases:

- Input: `count_element([1, 2, 2, 3, 2], 2)` → Output: `3`
- Input: `count_element(['a', 'b', 'a'], 'a')` → Output: `2`

72. Remove Duplicates

Write a function that removes duplicates from a list.

```
python
```

```
def remove_duplicates(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `remove_duplicates([1, 2, 2, 3, 1])` → Output: `[1, 2, 3]`
- Input: `remove_duplicates(['a', 'b', 'a'])` → Output: `['a', 'b']`

73. Find Index

Write a function that finds the index of the first occurrence of an element (return -1 if not found).

```
python
```

```
def find_index(lst, element):
    # Your code here
    pass
```

Test Cases:

- Input: `find_index([1, 2, 3, 4], 3)` → Output: `2`
- Input: `find_index([1, 2, 3], 5)` → Output: `-1`

74. Even Numbers Only

Write a function that returns a list of only even numbers from the input list.

```
python
```

```
def get_evens(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `get_evens([1, 2, 3, 4, 5, 6])` → Output: `[2, 4, 6]`
- Input: `get_evens([1, 3, 5])` → Output: `[]`

75. Odd Numbers Only

Write a function that returns a list of only odd numbers from the input list.

```
python
```

```
def get_odds(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `get_odds([1, 2, 3, 4, 5, 6])` → Output: `[1, 3, 5]`
- Input: `get_odds([2, 4, 6])` → Output: `[]`

76. Merge Lists

Write a function that merges two lists into one.

```
python
```

```
def merge_lists(lst1, lst2):
    # Your code here
    pass
```

Test Cases:

- Input: `merge_lists([1, 2], [3, 4])` → Output: `[1, 2, 3, 4]`
- Input: `merge_lists(['a'], ['b', 'c'])` → Output: `['a', 'b', 'c']`

77. List Contains

Write a function that checks if a list contains a specific element.

```
python
```

```
def list_contains(lst, element):
    # Your code here
    pass
```

Test Cases:

- Input: `list_contains([1, 2, 3], 2)` → Output: `True`
- Input: `list_contains([1, 2, 3], 5)` → Output: `False`

78. Second Largest

Write a function that finds the second largest number in a list.

```
python
```

```
def second_largest(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `second_largest([1, 5, 3, 9, 2])` → Output: `5`
- Input: `second_largest([10, 20, 30])` → Output: `20`

79. Positive Numbers Only

Write a function that returns a list of only positive numbers.

```
python
```

```
def get_positives(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `get_positives([1, -2, 3, -4, 5])` → Output: `[1, 3, 5]`
- Input: `get_positives([-1, -2, -3])` → Output: `[]`

80. Square All Elements

Write a function that returns a list with all elements squared.

```
python
```

```
def square_all(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `square_all([1, 2, 3])` → Output: `[1, 4, 9]`
- Input: `square_all([5, 10])` → Output: `[25, 100]`

81. Cumulative Sum

Write a function that returns the cumulative sum list.

```
python
```

```
def cumulative_sum(lst):  
    # Your code here  
    pass
```

Test Cases:

- Input: `cumulative_sum([1, 2, 3, 4])` → Output: `[1, 3, 6, 10]`
- Input: `cumulative_sum([5, 5, 5])` → Output: `[5, 10, 15]`

82. Flatten List

Write a function that flattens a 2D list into a 1D list.

```
python
```

```
def flatten(lst):  
    # Your code here  
    pass
```

Test Cases:

- Input: `flatten([[1, 2], [3, 4]])` → Output: `[1, 2, 3, 4]`
- Input: `flatten([[1], [2, 3], [4]])` → Output: `[1, 2, 3, 4]`

83. Common Elements

Write a function that returns common elements between two lists.

```
python
```

```
def common_elements(lst1, lst2):  
    # Your code here  
    pass
```

Test Cases:

- Input: `common_elements([1, 2, 3], [2, 3, 4])` → Output: `[2, 3]`
- Input: `common_elements(['a', 'b'], ['b', 'c'])` → Output: `['b']`

84. Rotate List

Write a function that rotates a list to the right by k positions.

```
python
```

```
def rotate_list(lst, k):  
    # Your code here  
    pass
```

Test Cases:

- Input: `rotate_list([1, 2, 3, 4, 5], 2)` → Output: `[4, 5, 1, 2, 3]`
- Input: `rotate_list([1, 2, 3], 1)` → Output: `[3, 1, 2]`

85. List Range Sum

Write a function that returns the sum of elements between two indices (inclusive).

```
python
```

```
def range_sum(lst, start, end):  
    # Your code here  
    pass
```

Test Cases:

- Input: `range_sum([1, 2, 3, 4, 5], 1, 3)` → Output: `9` ($2+3+4$)
- Input: `range_sum([10, 20, 30], 0, 2)` → Output: `60`

Dictionaries (Questions 86-95)

86. Create Dictionary

Write a function that creates a dictionary from two lists (keys and values).

```
python
```

```
def create_dict(keys, values):  
    # Your code here  
    pass
```

Test Cases:

- Input: `create_dict(['a', 'b', 'c'], [1, 2, 3])` → Output: `{'a': 1, 'b': 2, 'c': 3}`

87. Get Value

Write a function that returns the value for a key (or "Not Found" if key doesn't exist).

```
python

def get_value(d, key):
    # Your code here
    pass
```

Test Cases:

- Input: `get_value({'a': 1, 'b': 2}, 'a')` → Output: `1`
- Input: `get_value({'a': 1}, 'z')` → Output: `"Not Found"`

88. Count Keys

Write a function that returns the number of keys in a dictionary.

```
python

def count_keys(d):
    # Your code here
    pass
```

Test Cases:

- Input: `count_keys({'a': 1, 'b': 2, 'c': 3})` → Output: `3`
- Input: `count_keys({})` → Output: `0`

89. Merge Dictionaries

Write a function that merges two dictionaries.

```
python

def merge_dicts(d1, d2):
    # Your code here
    pass
```

Test Cases:

- Input: `merge_dicts({'a': 1}, {'b': 2})` → Output: `{'a': 1, 'b': 2}`

90. Invert Dictionary

Write a function that inverts a dictionary (swap keys and values).

```
python
```

```
def invert_dict(d):
    # Your code here
    pass
```

Test Cases:

- Input: `invert_dict({'a': 1, 'b': 2})` → Output: `{1: 'a', 2: 'b'}`

91. Key Exists

Write a function that checks if a key exists in a dictionary.

```
python
```

```
def key_exists(d, key):
    # Your code here
    pass
```

Test Cases:

- Input: `key_exists({'a': 1, 'b': 2}, 'a')` → Output: `True`
- Input: `key_exists({'a': 1}, 'z')` → Output: `False`

92. Get All Keys

Write a function that returns a list of all keys in a dictionary.

```
python
```

```
def get_keys(d):
    # Your code here
    pass
```

Test Cases:

- Input: `get_keys({'a': 1, 'b': 2, 'c': 3})` → Output: `['a', 'b', 'c']`

93. Get All Values

Write a function that returns a list of all values in a dictionary.

```
python
```

```
def get_values(d):
    # Your code here
    pass
```

Test Cases:

- Input: `get_values({'a': 1, 'b': 2, 'c': 3})` → Output: `[1, 2, 3]`

94. Filter by Value

Write a function that returns a dictionary with only entries where value > threshold.

```
python
```

```
def filter_dict(d, threshold):
    # Your code here
    pass
```

Test Cases:

- Input: `filter_dict({'a': 1, 'b': 5, 'c': 3}, 2)` → Output: `{'b': 5, 'c': 3}`

95. Sum of Values

Write a function that returns the sum of all values in a dictionary.

```
python
```

```
def sum_values(d):
    # Your code here
    pass
```

Test Cases:

- Input: `sum_values({'a': 1, 'b': 2, 'c': 3})` → Output: `6`

Mixed Challenges (Questions 96-100)

96. FizzBuzz

Write a function that returns a list of strings from 1 to n: "Fizz" for multiples of 3, "Buzz" for multiples of 5, "FizzBuzz" for both, or the number as a string.

```
python
```

```
def fizzbuzz(n):
    # Your code here
    pass
```

Test Cases:

- Input: `fizzbuzz(15)` → Output: `['1', '2', 'Fizz', '4', 'Buzz', 'Fizz', '7', '8', 'Fizz', 'Buzz', '11', 'Fizz', '13', '14', 'FizzBuzz']`

97. Password Validator

Write a function that validates a password (at least 8 chars, has digit, has uppercase, has lowercase).

```
python
```

```
def validate_password(password):
    # Your code here
    pass
```

Test Cases:

- Input: `validate_password("Abcd1234")` → Output: `True`
- Input: `validate_password("abc")` → Output: `False`
- Input: `validate_password("ABCD1234")` → Output: `False` (no lowercase)

98. Caesar Cipher

Write a function that encrypts a string using Caesar cipher with given shift.

```
python
```

```
def caesar_cipher(text, shift):
    # Your code here (only shift letters, keep others same)
    pass
```

Test Cases:

- Input: `caesar_cipher("abc", 1)` → Output: `"bcd"`
- Input: `caesar_cipher("xyz", 3)` → Output: `"abc"`

99. Most Frequent Element

Write a function that returns the most frequent element in a list.

```
python
```

```
def most_frequent(lst):
    # Your code here
    pass
```

Test Cases:

- Input: `most_frequent([1, 2, 2, 3, 2])` → Output: `2`
- Input: `most_frequent(['a', 'b', 'a', 'a'])` → Output: `'a'`

100. Matrix Addition

Write a function that adds two 2D matrices (represented as lists of lists).

```
python
```

```
def add_matrices(m1, m2):
    # Your code here
    pass
```

Test Cases:

- Input: `add_matrices([[1, 2], [3, 4]], [[5, 6], [7, 8]])` → Output: `[[6, 8], [10, 12]]`
- Input: `add_matrices([[1]], [[2]])` → Output: `[[3]]`

Tips for Practice:

1. Try to solve each problem without looking at built-in functions first
2. Test your solution with the provided test cases
3. Think about edge cases (empty inputs, negative numbers, etc.)
4. Try to optimize your solution after getting it working
5. Practice explaining your solution out loud

Good luck with your Python practice!