```
# 1.
def print_name_age(name, age):
    print(f"My name is {name} and I am {age} years old.")
print_name_age("Raman", 27)
```

```
My name is Raman and I am 27 years old.
```

```
# 2.
def add_numbers(a, b):
    print("Sum =", a + b)
add_numbers(10, 15)
```

```
Sum = 25
```

```
# 3.
def check_number(num):
    if num > 0: print("Positive")
    elif num < 0: print("Negative")
    else: print("Zero")
check_number(-8)
```

```
Negative
```

```
# 4.
def square(num):
    print("Square:", num ** 2)
square(7)
```

```
Square: 49
```

```
# 5.
def even_numbers():
    print("Even Numbers:", [i for i in range(1, 51) if i % 2 == 0])
even_numbers()
```

```
Even Numbers: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30,
32, 34, 36, 38, 40, 42, 44, 46, 48, 50]
```

```
# 6.
def multiplication_table(num):
    print(f"Multiplication table for {num}")
    for i in range(1, 11):
        print(f"{num} x {i} = {num * i}")
multiplication_table(9)
```

```
Multiplication table for 9
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
```

```
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90
```

```python
# 7.
def divisible_by_five(num):
    if num % 5 == 0:
        print(num, "is divisible by 5")
    else:
        print(num, "is not divisible by 5")
divisible_by_five(30)
```

```
30 is divisible by 5
```

```python
# 8.
def concatenate_strings(s1, s2):
    print("Concatenated string:", s1 + s2)
concatenate_strings("Hello ", "Python")
```

```
Concatenated string: Hello Python
```

```python
# 9.
def count_word(sentence, word):
    count = sentence.lower().split().count(word.lower())
    print(f"'{word}' appears {count} times")
count_word("Python is easy and Python is powerful", "Python")
```

```
'Python' appears 2 times
```

```python
# 10.
def string_length(s):
    length = 0
    for _ in s:
        length += 1
    print("Length:", length)
string_length("Developer")
```

```
Length: 9
```

```python
# 11.
def count_vowels(s):
    vowels = "aeiouAEIOU"
    count = 0
    for ch in s:
        if ch in vowels:
            count += 1
    print("Vowel Count:", count)
count_vowels("Programming")
```

```
Vowel Count: 3
```

```python
# 12.
def starts_with_vowel(s):
    if s:
        print("Starts with vowel:", s[0].lower() in 'aeiou')
    else:
        print("Empty string")
starts_with_vowel("Apple")
```

```
Starts with vowel: True
```

```python
# 13.
def remove_spaces(s):
    print("Without spaces:", "".join(ch for ch in s if ch != " "))
remove_spaces("Python is Fun")
```

```
Without spaces: PythonisFun
```

```python
# 14.
def replace_vowels(s):
    result = ""
    for ch in s:
        result += "*" if ch.lower() in "aeiou" else ch
    print("After replacing vowels:", result)
replace_vowels("Education")
```

```
After replacing vowels: *d*c*t**n
```

```python
# 15.
def count_case(s):
    upper = 0
    lower = 0
    for ch in s:
        if ch.isupper(): upper += 1
        elif ch.islower(): lower += 1
    print(f"Uppercase: {upper}, Lowercase: {lower}")
count_case("HelloPython")
```

```
Uppercase: 2, Lowercase: 9
```

```python
# 16.
def reverse_string(s):
    print("Reversed:", s[::-1])
reverse_string("Laptop")
```

```
Reversed: potpaL
```

```python
# 17.
def is_palindrome(s):
    print("Palindrome:", s == s[::-1])
is_palindrome("madam")
```

```
Palindrome: True
```

```python
# 18.
def word_count(sentence):
    print("Word count:", len(sentence.split()))
word_count("This is an example sentence")
```

```
Word count: 5
```

```python
# 19.
def most_repeated_char(s):
    max_char = ''
    max_count = 0
    for ch in set(s):
        count = s.count(ch)
        if count > max_count:
            max_count = count
            max_char = ch
    print("Most repeated character:", max_char)
most_repeated_char("success")
```

```
Most repeated character: s
```

```python
# 20.
def to_upper(s):
    result = ""
    for ch in s:
        if 'a' <= ch <= 'z':
            result += chr(ord(ch) - 32)
        else:
            result += ch
    print("Uppercase version:", result)
to_upper("python")
```

```
Uppercase version: PYTHON
```

```python
# 21.
def greatest(lst):
    print("Greatest:", max(lst))
greatest([10, 55, 3, 42])
```

```
Greatest: 55
```

```python
# 22.
def smallest(lst):
    print("Smallest:", min(lst))
smallest([10, 55, 3, 42])
```

```
Smallest: 3
```

```python
# 23.
def create_list():
    lst = [2, 4, 6, 8, 10]
    print("List Created:", lst)
create_list()
```

```
List Created: [2, 4, 6, 8, 10]
```

```
# 24.
def remove_duplicates(lst):
    unique = []
    for item in lst:
        if item not in unique:
            unique.append(item)
    print("After removing duplicates:", unique)
remove_duplicates([1, 2, 2, 4, 4, 5])
```

```
After removing duplicates: [1, 2, 4, 5]
```

```
# 25.
def count_even_odd(lst):
    even = 0
    odd = 0
    for i in lst:
        if i % 2 == 0:
            even += 1
        else:
            odd += 1
    print(f"Evens: {even}, Odds: {odd}")
count_even_odd([1, 2, 3, 4, 5, 6])
```

```
Evens: 3, Odds: 3
```

```
# 26.
def sum_list(lst):
    total = 0
    for num in lst:
        total += num
    print("Sum:", total)
sum_list([5, 10, 15])
```

```
Sum: 30
```

```
# 27.
def sort_list(lst):
    for i in range(len(lst)):
        for j in range(i + 1, len(lst)):
            if lst[i] > lst[j]:
                lst[i], lst[j] = lst[j], lst[i]
    print("Sorted:", lst)
sort_list([7, 3, 8, 1, 6])
```

```
Sorted: [1, 3, 6, 7, 8]
```

```python
# 28.
def smallest_string(lst):
    smallest = lst[0]
    for s in lst:
        if len(s) < len(smallest):
            smallest = s
    print("Smallest string:", smallest)
smallest_string(["hi", "hello", "hey"])
```

```
Smallest string: hi
```

```python
# 29.
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    print("Factorial:", result)
factorial(6)
```

```
Factorial: 720
```

```python
# 30.
def is_prime(n):
    if n < 2:
        print("Not Prime")
        return
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            print("Not Prime")
            return
    print("Prime Number")
is_prime(37)
```

```
Prime Number
```

```python
# 31.
def primes_1_to_100():
    primes = []
    for i in range(2, 101):
        for j in range(2, int(i ** 0.5) + 1):
            if i % j == 0:
                break
        else:
            primes.append(i)
    print("Prime numbers 1–100:", primes)
primes_1_to_100()
```

```
Prime numbers 1–100: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

```python
# 32.
def sum_of_digits(n):
    total = 0
    for ch in str(n):
        total += int(ch)
    print("Sum of digits:", total)
sum_of_digits(987)
```

```
Sum of digits: 24
```

```python
# 33.
def fibonacci(n):
    a, b = 0, 1
    result = []
    for _ in range(n):
        result.append(a)
        a, b = b, a + b
    print("Fibonacci:", result)
fibonacci(10)
```

```
Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

```python
# 34.
def is_armstrong(n):
    digits = str(n)
    power = len(digits)
    total = 0
    for d in digits:
        total += int(d) ** power
    print("Armstrong number:", total == n)
is_armstrong(153)
```

```
Armstrong number: True
```

```python
#35
def create_text_file():
    try:
        file = open("myfile.txt", "w")  # create/write mode
        file.write("Hello, Python")
        file.close()
        print("File created and written successfully.")
    except Exception as e:
        print("Error:", e)

create_text_file()
```

```
File created and written successfully.
```

```python
#36

def read_text_file():
    try:
        file = open("myfile.txt", "r")
        content = file.read()
        file.close()
        print("File content:")
        print(content)
    except FileNotFoundError:
        print("File not found. Please create it first.")
    except Exception as e:
        print("Error:", e)

read_text_file()
```

```
File content:
Hello, Python
```

```python
#37

def count_file_statistics():
    try:
        with open("myfile.txt", "r") as file:
            text = file.read()
            lines = text.splitlines()
            words = text.split()
            chars = len(text)
            print(f"Lines: {len(lines)}")
            print(f"Words: {len(words)}")
            print(f"Characters: {chars}")
    except FileNotFoundError:
        print("File not found.")
    except Exception as e:
        print("Error:", e)

count_file_statistics()
```

```
Lines: 1
Words: 2
Characters: 13
```

```python
#38
def append_to_file(new_text):
    try:
        with open("myfile.txt", "a") as file:
            file.write("\n" + new_text)
        print("Text appended successfully.")
    except FileNotFoundError:
        print("File not found.")
    except Exception as e:
        print("Error:", e)
```

```
append_to_file("This line is newly added.")
read_text_file()
```

```
Text appended successfully.
File content:
Hello, Python
This line is newly added.
```

```
# 39.
def guess_number():
    number = 45
    guess = 43
    print(f"Guessed {guess}, actual {number}")
    print("Too low!" if guess < number else "Correct!" if guess ==
number else "Too high!")
guess_number()
```

```
Guessed 43, actual 45
Too low!
```

```
# 40.
def roll_two_dice():
    d1, d2 = 3, 5
    print("Total:", d1 + d2)
roll_two_dice()

# Recursive sum
def recursive_sum(n):
    if n == 0:
        return 0
    return n + recursive_sum(n - 1)
print("Recursive sum till 5:", recursive_sum(5))
```

```
Total: 8
Recursive sum till 5: 15
```

```
# 41.
def are_anagrams(a, b):
    print("Anagram check:", sorted(a) == sorted(b))
are_anagrams("listen", "silent")
```

```
Anagram check: True
```

```
# 42.
def sort_sentence(sentence):
    words = sentence.split()
    for i in range(len(words)):
        for j in range(i + 1, len(words)):
            if words[i].lower() > words[j].lower():
                words[i], words[j] = words[j], words[i]
    print("Alphabetical:", " ".join(words))
sort_sentence("Python is easy")
```

```
Alphabetical: easy is Python
```

```
# 43.
def even_squares():
    evens = [i * i for i in range(1, 51) if i % 2 == 0]
    print("Even Squares:", evens)
even_squares()
```

```
Even Squares: [4, 16, 36, 64, 100, 144, 196, 256, 324, 400, 484, 576,
676, 784, 900, 1024, 1156, 1296, 1444, 1600, 1764, 1936, 2116, 2304,
2500]
```

```
# 44.
def unique_list(lst):
    unique = []
    for x in lst:
        if x not in unique:
            unique.append(x)
    print("Unique list:", unique)
unique_list([1, 1, 2, 3, 3, 4])
```

```
Unique list: [1, 2, 3, 4]
```

```
# 45.
cube = lambda n: n ** 3
print("Cube of 6:", cube(6))
```

```
Cube of 6: 216
```

```
# 46.
def shuffle_list(lst):
    n = len(lst)
    for i in range(0, n - 1, 2):
        lst[i], lst[i + 1] = lst[i + 1], lst[i]
    print("Shuffled (manual swap):", lst)
shuffle_list([1, 2, 3, 4, 5, 6])
```

```
Shuffled (manual swap): [2, 1, 4, 3, 6, 5]
```

```
# 47.
def random_password():
    password = "Code1234"
    print("Generated Password:", password)
random_password()
```

```
Generated Password: Code1234
```

```
# 48.
def count_above_average(lst):
```

```python
    avg = sum(lst)/len(lst)
    count = 0
    for value in lst:
        if value > avg:
            count += 1
    print(f"Above average elements: {count}")
count_above_average([10, 20, 30, 40, 50, 60])
```

```
Above average elements: 3
```

```python
# 49.
def simple_calculator(a, b, op):
    if op == '+': print("Sum:", a + b)
    elif op == '-': print("Difference:", a - b)
    elif op == '*': print("Product:", a * b)
    elif op == '/': print("Quotient:", a / b)
simple_calculator(9, 3, '*')
```

```
Product: 27
```

```python
# 50.
def greatest_substring(sentence):
    words = sentence.split()
    longest = words[0]
    for word in words:
        if len(word) > len(longest):
            longest = word
    print("Longest word:", longest)
greatest_substring("Python makes backend development easier")
```

```
Longest word: development
```