```
D:\Work\VSCode\Pyt
Q01_HelloWorld.py
Hello World
```

```
D:\Work\VSCode\Pyt
Q02_PrintName.py
Smart Brains
```

```
D:\Work\VSCode\Python_Codes\src>py
Q03_CheckValidIdentifiers.py
Enter an identifier to check: asd
'asd' is a valid identifier.
```

```
D:\Work\VSCode\Python_Codes\src>python Q04_CheckIr
Q04_CheckInvalidIdentifiers.py
Enter an identifier to check if it's invalid: asd
'asd' is a valid identifier.
```

```
D:\Work\VSCode\Python_Cod
Q05_SingleLineComment.py
Hello, World!
```

```
D:\Work\VSCode\Python_Coc
Q06_MultilineComment.py
Multiline comment
```

```
D:\Work\VSCode\Python_Co
Q07_DocStringComment.py
Docstring comment.
```

```
D:\Work\VSCode\Python_Codes\sr
Q08_SingleFunction.py
Hello from a single function!
```

```
D:\Work\VSCode\Python_Codes\src>py
Q09_VariableTypes.py
Integer Variables:
age = 25
Type of age is: <class 'int'>

Float Variables:
price = 45.99
Type of price is: <class 'float'>
```

```
D:\Work\VSCode\Python_Coc
Q10_DataTypes.py
=== String Examples ===
Name: John Doe

=== Boolean Examples ===
is_python: True

=== Tuple Examples ===
Coordinates: (10, 20)
X coordinate: 10
Y coordinate: 20
```

```
D:\Work\VSCode\Python_Codes\src>python Q11_ListSetDict.p
Q11_ListSetDict.py
List Example:
Fruits:  ['apple', 'banana', 'orange']

Set Example:
Colors:  {'green', 'blue', 'red'}

Dictionary Example:
Person: {'name': 'John', 'age': 25, 'city': 'New York'}
```

```
D:\Work\VSCode\Python_Codes\sr
Q12_GettingUserName.py
Enter Username: SmartBrains
Username: SmartBrains
```

```
D:\Work\VSCode\Pyth
Q13_Add10.py
Enter a number: 10
10 + 10 = 20
```

```
D:\Work\VSCode\Python_Codes\src>
Q14_MultipleInputs.py
Enter your name: SmartBrain
Enter your age: 20
Enter your city: Delhi
Enter your favorite hobby: Code
Enter your height in feet: 5

=== User Information ===

Age: 20 years
City: Delhi
Favorite Hobby: Code
Height: 5.00 meters
```

```
D:\Work\VSCode\Python_Codes\src>python Q15
Q15_BasicMathOperations.py
Enter two numbers to perform basic math op
Enter first number: 2
Enter second number: 3

Mathematical Operations with 2.0 and 3.0:
Addition: 2.0 + 3.0 = 5.0
Subtraction: 2.0 - 3.0 = -1.0
Multiplication: 2.0 * 3.0 = 6.0
Division: 2.0 / 3.0 = 0.67
Floor Division: 2.0 // 3.0 = 0.0
Modulus: 2.0 % 3.0 = 2.0
Power: 2.0 ** 3.0 = 8.0
```

```
D:\Work\VSCode\Python_Codes\src>python Q
Q16_MathModule.py
=== Mathematical Constants ===
Pi (π): 3.141592653589793
Euler's number (e): 2.718281828459045
Tau (τ): 6.283185307179586

=== Basic Mathematical Functions ===
Square root of 16: 4.0
Factorial of 5: 120
GCD of 48 and 60: 12
Absolute value of -7.8: 7.8

=== Trigonometric Functions ===
Angle: 45 degrees = 0.7854 radians
sin(45°): 0.7071
cos(45°): 0.7071
tan(45°): 1.0000

=== Logarithmic Functions ===
Natural logarithm of 100: 4.6052
Base-10 logarithm of 100: 2.0000
Base-2 logarithm of 100: 6.6439

=== Power and Exponential Functions ===
2 raised to power 3: 8.0
e raised to power 2: 7.3891

=== Ceiling and Floor ===
Ceiling of 7.6: 8
Floor of 7.6: 7
```

```
D:\Work\VSCode\Python_Codes\src
Q17_ByteToBinary.py
===BYTE EXAMPLE===
b'ABCDE'
65
66
6566676869
===BYTE EXAMPLE===
bytearray(b'ABCDE')
bytearray(b'aBCDE')
bytearray(b'aBCDEF')
b'aBCDEF'

===Memory view example===
<memory at 0x000001F07EDB5E40>


65
b'BCD'


bytearray(b'aBCDE')
bytearray(b'aBCDE')
```

```
D:\Work\VSCode\Python_Codes\src>python Q18_ListOperati
Q18_ListOperations.py
Enter how many numbers you want to add to the list: 2
Enter number 1: 5
Enter number 2: 6

Results:
Original list: [5.0, 6.0]
Sum of numbers: 11.0
Sorted list: [5.0, 6.0]
```

```
D:\Work\VSCode\Python_Codes\src>p
Q19_SwapNumbers.py
Enter first number: 20
Enter second number: 40
Before swapping: a = 20, b = 40
After swapping: a = 40, b = 20
```

```
D:\Work\VSCode\Python_Codes\src
Q20_CheckPositiveOrNegative.py
Enter a number: 34
The number is positive
```

```
D:\Work\VSCode\Pytho
Q21_OddEven.py
Enter a number: 33
The number is odd
```

```
D:\Work\VSCode\Python_Cod
Q22_LeapYear.py
Enter a year: 2025
2025 is not a leap year
```

```
D:\Work\VSCode\Python_Codes\src>
Q23_LastDigit.py
Enter an integer number: 123123
The last digit of 123123 is: 3
```

```
D:\Work\VSCode\Python_Codes\src
Q24_StudentPercentage.py
Enter marks for subject 1: 90
Enter marks for subject 2: 91
Enter marks for subject 3: 92
Enter marks for subject 4: 93
Enter marks for subject 5: 99
Total marks: 465.0
Percentage: 93.0%
```

```
D:\Work\VSCode\Python_Codes\sr
Q25_StudentGrade.py
Enter marks for subject 1: 90
Enter marks for subject 2: 91
Enter marks for subject 3: 92
Enter marks for subject 4: 93
Enter marks for subject 5: 99
Total marks: 465.0
Percentage: 93.0%
Grade: A+
```

```
D:\Work\VSCode\Python_Codes\src>
Q26_SaveReportToFile.py
Enter your name: SmartBrain
Enter your age: 20
Enter your city: delhi
Report saved to user_report.txt
```

```
User Report
-----------
Name: SmartBrain
Age: 20
City: delhi
```

```
D:\Work\VSCode\Python_Codes\
Q27_CheckVowel.py
Enter a single character: s
's' is not a vowel
```

```
D:\Work\VSCode\Python_Codes\src>p
Q28_DivisibleBy2And3.py
Enter a number: 12
12 is divisible by both 2 and 3
```

```
D:\Work\VSCode\Python_Codes\src
Q29_LargestOfThree.py
Enter first number: 123123
Enter second number: 123123
Enter third number: 123123
The largest number is: 123123.0
```

```
D:\Work\VSCode\Python_Codes\src>
Q30_LoginPassword.py
Enter username: SmartBrains
Enter password: SmartBrains@05
Invalid username or password
```

```
D:\Work\VSCode\Python_Codes\s
Q31_SeasonBasedOnMonth.py
Enter month number (1-12): 3
Season: Spring
```

```
D:\Work\VSCode\Python
Q32_Factorial.py
Enter a number: 4
Factorial of 4 is 24
```

```
D:\Work\VSCode\Python_Codes\src>python Q104_
Q104_StringOperations.py
Original String: Hello World

=== String Operations ===
1. Uppercase: HELLO WORLD
2. Lowercase: hello world
3. Length: 11
4. Replace: Hello Python
5. Split: ['Hello', 'World']
6. Find 'World': 6
7. Count 'l': 3
8. Starts with 'Hello': True
9. Ends with 'World': True
10. Concatenation: Hello World from Python
```

```
D:\Work\VSCode\Python_
Q33_TableFrom2To10.py

Table of 2:
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

Table of 3:
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

Table of 4:
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40

Table of 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

Table of 6:
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60

Table of 7:
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70

Table of 8:
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80

Table of 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

Table of 10:
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100
```

```
D:\Work\VSCode\Python_Codes\src>pyth
Q36_PrimeInRange.py
Enter start of range: 10
Enter end of range: 20

Prime numbers between 10 and 20:
11 13 17 19
D:\Work\VSCode\Python_Codes\src>pyth
Q37_ATMPin.py
Attempt 1: Enter your ATM PIN: 4321
Incorrect PIN!
Attempt 2: Enter your ATM PIN: 1234
PIN accepted! Access granted.

D:\Work\VSCode\Python_Codes\src>pyth
Q38_DayOfWeek.py
Enter a number (1-7): 3
Wednesday

D:\Work\VSCode\Python_Codes\src>pyth
Q39_CheckArmstrong.py
Enter a number: 157
157 is not an Armstrong number

D:\Work\VSCode\Python_Codes\src>pyth
Q40_ArmstrongFrom1To1000.py
Armstrong numbers from 1 to 1000:
1 2 3 4 5 6 7 8 9 153 370 371 407
D:\Work\VSCode\Python_Codes\src>
```

```
D:\Work\VSCode\Python_Codes\src>python Q41_SumOfNatu
Q41_SumOfNaturalNumbers.py
Enter a number: 2
Sum of first 2 natural numbers is 3

D:\Work\VSCode\Python_Codes\src>python Q42_Fibonacci
Q42_FibonacciSequence.py
How many Fibonacci numbers do you want? 3
First 3 Fibonacci numbers:
0 1 1
D:\Work\VSCode\Python_Codes\src>python Q43_StringOpe
Q43_StringOperations.py
Original String: Hello World

--- String Operations ---

1. Uppercase: HELLO WORLD
2. Lowercase: hello world
3. Length: 11
4. Replace 'World' with 'Python': Hello Python
5. Split into words: ['Hello', 'World']
6. Find 'World': 6
7. Count 'o': 2
8. Starts with 'Hello': True
9. Ends with 'World': True
10. Concatenation: Hello World from Python

D:\Work\VSCode\Python_Codes\src>python Q44_ListOpera
Q44_ListOperations.py
Original List: [10, 20, 30, 40, 50]

--- List Operations ---

1. After append(60): [10, 20, 30, 40, 50, 60]
2. After insert(2, 25): [10, 20, 25, 30, 40, 50, 60]
3. After remove(25): [10, 20, 30, 40, 50, 60]
ed: 60
```

```
D:\Work\VSCode\Python_Codes\src>p
Q34_CelsiusToFahrenheit.py
Enter temperature in Celsius: 34
34.0°C = 93.2°F
```

```
D:\Work\VSCode\Python_Co
Q35_CheckPrimeNumber.py
Enter a number: 19
19 is a prime number
```

```
D:\Work\VSCode\Python_Codes\src>python Q48_StringLengt
Q48_StringLengthWithoutLen.py
Enter a string: SmartBrains
Length of 'SmartBrains' is 11

D:\Work\VSCode\Python_Codes\src>python Q49_CountWordO
Q49_CountWordOccurrence.py
Enter a sentence: SmartBrains is 2nd runner up in cc1
Enter the word to count: 10
The word '10' appears 0 times in the sentence
```

```
ıon_Codes\src>py
ing.py
artBrains
OT a palindrome
```

```
_Codes\src>
Q47_CommonCharacters.py
Enter first string: SmartBrains
Enter second string: BrainsOG

Common characters:
a r B i n s
```

```
D:\Work\VSCode\Python_Codes\src>python Q4
Q46_StringTraversing.py
Enter a string: SmartBrains

Traversing string character by character:

Character: S
Character: m
Character: a
Character: r
Character: t
Character: B
Character: r
Character: a
Character: i
Character: n
Character: s

Traversing with index:
Index 0: S
Index 1: m
Index 2: a
Index 3: r
Index 4: t
Index 5: B
Index 6: r
Index 7: a
Index 8: i
Index 9: n
Index 10: s
```

```
D:\Work\VSCode\Python_Codes\src>python Q53_MapFilter.py
Q53_MapFilter.py
Original List: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

=== map() Function ===

1. Squared numbers: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
2. Doubled numbers: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
3. Add 10 to each: [11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

=== filter() Function ===

1. Even numbers: [2, 4, 6, 8, 10]
2. Odd numbers: [1, 3, 5, 7, 9]
3. Numbers > 5: [6, 7, 8, 9, 10]

=== Combining map() and filter() ===

Even numbers squared: [4, 16, 36, 64, 100]
```

```
D:\Work\VSCode\Python_Codes\src>python Q55_SetOperations.p
Q55_SetOperations.py
Set 1: {1, 2, 3, 4, 5}
Set 2: {4, 5, 6, 7, 8}

=== Set Operations ===

1. Union (set1 | set2): {1, 2, 3, 4, 5, 6, 7, 8}
   Union using union(): {1, 2, 3, 4, 5, 6, 7, 8}

2. Intersection (set1 & set2): {4, 5}
   Intersection using intersection(): {4, 5}

3. Difference (set1 - set2): {1, 2, 3}
   Difference using difference(): {1, 2, 3}

4. Symmetric Difference (set1 ^ set2): {1, 2, 3, 6, 7, 8}
   Using symmetric_difference(): {1, 2, 3, 6, 7, 8}

5. After adding 6 to set1: {1, 2, 3, 4, 5, 6}
6. After removing 6: {1, 2, 3, 4, 5}
7. After discard(10): {1, 2, 3, 4, 5}

8. Is 3 in set1? True
9. Length of set1: 5
10. After clear(): set()
```

```
D:\Work\VSCode\Python_Codes\src>python Q50_TupleOpera
Q50_TupleOperations.py
Original Tuple: (10, 20, 30, 40, 50, 20)

--- Tuple Operations ---

1. Element at index 2: 30
2. Slice [1:4]: (20, 30, 40)
3. Length: 6
4. Count of 20: 2
5. Index of 30: 2
6. Is 40 in tuple? True
7. Maximum value: 50
8. Minimum value: 10
9. Sum of elements: 170
10. Concatenation: (10, 20, 30, 40, 50, 20, 60, 70)
11. Convert to list: [10, 20, 30, 40, 50, 20]
```

```
D:\Work\VSCode\Python_Codes\src>python Q51_DictionaryOperations.py
Q51_DictionaryOperations.py
Original Dictionary: {'name': 'Kaivalaya', 'roll': 205, 'branch': 'BCA', 'marks': 85}

--- Dictionary Operations ---

1. Name: Kaivalaya
2. Branch: BCA
3. After adding city: {'name': 'Kaivalaya', 'roll': 205, 'branch': 'BCA', 'marks': 85, 'city': 'Delhi'}
4. After updating marks: {'name': 'Kaivalaya', 'roll': 205, 'branch': 'BCA', 'marks': 90, 'city': 'Delhi'}
5. Keys: dict_keys(['name', 'roll', 'branch', 'marks', 'city'])
6. Values: dict_values(['Kaivalaya', 205, 'BCA', 90, 'Delhi'])
7. Items: dict_items([('name', 'Kaivalaya'), ('roll', 205), ('branch', 'BCA'), ('marks', 90), ('city', 'Delhi')])
8. Is 'name' in dict? True
9. After pop('city'): {'name': 'Kaivalaya', 'roll': 205, 'branch': 'BCA', 'marks': 90} | Removed: Delhi
10. Length: 4
11. Copied dictionary: {'name': 'Kaivalaya', 'roll': 205, 'branch': 'BCA', 'marks': 90}
12. After clear(): {}
```

```
D:\Work\VSCode\Python_Codes\src>python Q52_LenDelRemoveRang
Q52_LenDelRemoveRange.py
=== List Operations ===

Original List: [10, 20, 30, 40, 50]
1. Length of list: 5
2. After remove(30): [10, 20, 40, 50]
3. After del my_list[1]: [10, 40, 50]
4. List from range(1, 11): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

=== Tuple Operations ===

Original Tuple: (5, 10, 15, 20, 25)
1. Length of tuple: 5
2. Before del: (1, 2, 3)
   Tuple deleted successfully
3. Tuple from range(5, 16, 2): (5, 7, 9, 11, 13, 15)
4. Accessing tuple elements using range:
   Index 0: 5
   Index 1: 10
   Index 2: 15
   Index 3: 20
   Index 4: 25
```

```
D:\Work\VSCode\Python_Codes\src
Q56_YieldIterator.py
Countdown from 5:
5 4 3 2 1

Countdown from 10:
10 9 8 7 6 5 4 3 2 1

Custom Range from 1 to 10:
1 2 3 4 5 6 7 8 9 10

Even numbers up to 20:
0 2 4 6 8 10 12 14 16 18 20
```

```
D:\Work\VSCode\Python_Codes\src>python Q54_R
Q54_RangeFunctions.py
=== Different Range Functions ===

1. range(10):
0 1 2 3 4 5 6 7 8 9

2. range(5, 15):
5 6 7 8 9 10 11 12 13 14

3. range(0, 20, 2) - Even numbers:
0 2 4 6 8 10 12 14 16 18

4. range(10, 0, -1) - Countdown:
10 9 8 7 6 5 4 3 2 1

5. range(-5, 5):
-5 -4 -3 -2 -1 0 1 2 3 4

6. List from range(1, 11):
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

7. Tuple from range(5, 50, 5):
(5, 10, 15, 20, 25, 30, 35, 40, 45)

8. Length of range(1, 100): 99

9. Element at index 5 in range(10, 50): 15
```

```
D:\Work\VSCode\Python_Codes\src>python Q59_D
Q59_DifferentGenerators.py
=== Generator 1: Number Generator ===

Numbers 1 to 10:
1 2 3 4 5 6 7 8 9 10

=== Generator 2: Even Number Generator ===

Even numbers up to 20:
0 2 4 6 8 10 12 14 16 18 20

=== Generator 3: Factorial Generator ===

Factorials 1 to 5:
1 2 6 24 120

=== Generator 4: Power Generator ===

Powers of 2 (up to 2^8):
1 2 4 8 16 32 64 128 256

=== Generator 5: String Generator ===

Characters in 'Python':
P y t h o n

=== Generator 6: List Element Generator ===

Fruits:
Apple Banana Cherry Date
```

```
D:\Work\VSCode\Python_Codes\src>python Q61_Cou
Q61_CountVowelsConsonantsBlanks.py
Enter a string: SmartBrains

Results:
Total Vowels: 3
Total Consonants: 8
Total Blanks: 0

Total Characters: 11
Alphabets: 11

D:\Work\VSCode\Python_Codes\src>python Q62_Func
Q62_FunctionWithWithoutArgs.py
=== Function without arguments ===
Hello! Welcome to Python programming

=== Function with one argument ===
Hello Kaivalaya! Welcome to Python programming

=== Function with multiple arguments ===
Sum of 10 and 20 is 30
Sum of 50 and 75 is 125

=== Function with default argument ===
Good Morning, Kaivalaya!
Good Evening, Kaivalaya!

D:\Work\VSCode\Python_Codes\src>python Q63_LCM
Q63_LCMOfTwoNumbers.py
Enter first number: 20
Enter second number: 2

LCM of 20 and 2 is 20
```

```
D:\Work\VSCode\Python_Codes\src>python
Q57_GeneratorYield.py
Fibonacci sequence (first 10 numbers):
0 1 1 2 3 5 8 13 21 34

Squares from 1 to 10:
1 4 9 16 25 36 49 64 81 100

Even numbers up to 20:
0 2 4 6 8 10 12 14 16 18 20

Countdown from 5:
5 4 3 2 1

Using next() with generator:
First: 1
Second: 4
Third: 9
```

```
D:\Work\VSCode\Python_Codes\src>python Q60_Cr
Q60_CreateIterators.py
=== Iterator 1: Custom Counter Iterator ===

Counter from 1 to 10:
1 2 3 4 5 6 7 8 9 10

=== Iterator 2: Reverse Iterator ===

Reverse of 'Python':
n o h t y P

=== Iterator 3: Even Numbers Iterator ===

Even numbers up to 20:
0 2 4 6 8 10 12 14 16 18 20

=== Iterator 4: Fibonacci Iterator ===

First 10 Fibonacci numbers:
0 1 1 2 3 5 8 13 21 34
```

```
D:\Work\VSCode\Python_Codes\sr
Q65_FunctionNoReturn.py
Welcome to Python Programming!
Learning functions is fun!

Table of 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

--- Student Information ---
Name: Kaivalaya
Roll Number: 205
Marks: 85

=== Greeting 3 times ===
Hello Kaivalaya!
Hello Kaivalaya!
Hello Kaivalaya!
```

```
D:\Work\VSCode\Python_Codes\src>python Q58_ComprehensionWithCondition.py
Q58_ComprehensionWithCondition.py
=== List Comprehension ===

1. Squares of 1-10: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
2. Even numbers 1-20: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
3. Odd numbers 1-20: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
4. Even/Odd labels: ['Odd', 'Even', 'Odd', 'Even', 'Odd', 'Even', 'Odd', 'Even', 'Odd', 'Even']
5. Multiples of 3: [3, 6, 9, 12, 15, 18, 21, 24, 27, 30]

=== Set Comprehension ===

1. Squares set: {64, 1, 4, 36, 100, 9, 16, 49, 81, 25}
2. Even numbers set: {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}
3. Unique numbers: {1, 2, 3, 4, 5}
4. Unique characters in 'programming': {'r', 'm', 'g', 'i', 'p', 'n', 'a', 'o'}

=== Dictionary Comprehension ===

1. Squares dictionary: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
2. Even squares dict: {2: 4, 4: 16, 6: 36, 8: 64, 10: 100}
```

```
D:\Work\VSCode\Python_Codes'
Q64_LCMOfMultipleNumbers.py
How many numbers? 3
Enter the numbers:
Number 1: 3
Number 2: 2
Number 3: 1

LCM of [3, 2, 1] is 6
```

```
D:\Work\VSCode\Python_Codes\src>py
Q66_FunctionReturningValues.py
=== Square Function ===
Square of 5 is 25

=== Area Calculation ===
Area of rectangle (10 x 5) is 50

=== Circle Calculations ===
Circle with radius 7:
Area: 153.94
Circumference: 43.98

=== Even/Odd Check ===
Is 10 even? True
Is 7 even? False

=== Grade System ===
Marks 95: Grade A+
Marks 75: Grade B
Marks 55: Grade F
```

```
D:\Work\VSCode\Python_Codes\src>pyt
Q67_BankInterestCalculator.py
Enter customer name: Raunak
Enter account number: 101
Enter principal amount (₹): 100
Enter interest rate (%): 7
Enter time period (years): 2

==================================
         BANK INTEREST CALCULATOR
==================================
Customer Name: Raunak
Account Number: 101
Principal Amount: ₹100.0
==================================

--- Simple Interest ---
Interest: ₹14.00
Total Amount: ₹114.00

--- Compound Interest ---
Interest: ₹14.49
Total Amount: ₹114.49

Difference: ₹0.49
```

```
D:\Work\VSCode\Python_Codes\src>python
Q68_PrintAndReturn.py
Enter student name: Smart
Enter roll number: 101010
Enter marks obtained: 100
Enter total marks: 100

--- Calculating Results ---

Marks Obtained: 100.0/100.0
Percentage: 100.00%
Grade: A+

====================================
        STUDENT REPORT CARD
====================================
Name: Smart
Roll Number: 101010
Percentage: 100.00%
Grade: A+
====================================
Status: PASS - Excellent Performance!
```

```
D:\Work\VSCode\Python_Cod
Q69_HCFOfTwoNumbers.py
Enter first number: 123
Enter second number: 234


HCF of 123 and 234 is 3
HCF using recursion: 3
```

```
D:\Work\VSCode\Python_Codes\src>py
Q71_ASCIIValue.py
=== ASCII Value Finder ===

Enter a character: S

ASCII value of 'S' is 83

=== Reverse Conversion ===
Character for ASCII 83 is 'S'

=== Common ASCII Values ===
'A' = 65
'Z' = 90
'a' = 97
'z' = 122
'0' = 48
'9' = 57
' ' = 32
'@' = 64

=== ASCII Range ===
Uppercase A-Z: 65 to 90
Lowercase a-z: 97 to 122
Digits 0-9: 48 to 57
```

```
D:\Work\VSCode\Python_Codes\src
Q103_FileOperations.py
Data written to file

File content:
Alice,85
Bob,90
Charlie,78


=== Search Operation ===
Enter name to search: Alice
Found: Alice,85
```

```
D:\Work\VSCode\Python_Codes\src
Q74_FactorialRecursive.py
Enter a number: 3

Factorial of 3 is 6

Calculation:
3! = 3 × 2 × 1 = 6

Factorials from 1 to 3:
1! = 1
2! = 2
3! = 6
```

```
D:\Work\VSCode\Python_Codes\src>python Q
Q72_FibonacciRecursive.py
How many Fibonacci terms do you want? 5
Fibonacci sequence with 5 terms:
0 1 1 2 3

Detailed view:
F(0) = 0
F(1) = 1
F(2) = 1
F(3) = 2
F(4) = 3
```

```
D:\Work\VSCode\Python_Codes\s
Q73_DisplayCalendar.py
=== Calendar Program ===

1. Display Month Calendar
2. Display Year Calendar
3. Get Day Name
4. Check Leap Year

Enter your choice (1-4): 3
Enter year: 2
Enter month (1-12): 3
Enter day: 25

25/3/2 is a Monday
```

```
D:\Work\VSCode\Python_Codes\src>python Q76_Key
Q76_KeywordParameters.py
=== Example 1: Using keyword parameters (order

Name: Kaivalaya
Roll Number: 205
Branch: BCA
Marks: 85

=== Example 2: Different order with keywords =

Name: Alice
Roll Number: 101
Branch: CSE
Marks: 90

=== Example 3: Mixed positional and keyword ==

Name: Bob
Roll Number: 102
Branch: IT
Marks: 88

=== Example 4: Default keyword parameters ===


Title: Python Programming
Author: John Smith
Price: ₹0
Year: 2025

Title: Data Science
Author: Unknown
Price: ₹599
Year: 2025

Title: Unknown
Author: Unknown
Price: ₹0
Year: 2025

=== Example 5: Calculate percentage ===

Percentage 1: 84.33%
Percentage 2: 75.00%

=== Example 6: Greeting with keywords ===

Hello, Kaivalaya! Good Morning!
Hi, Alice! Good Evening!
Hey, Bob! Good Night!
```

```
D:\Work\VSCode\Python_Codes\src>python Q75
Q75_PositionalParameters.py
=== Example 1: Student Information ===

Name: Kaivalaya Dua
Roll Number: 205
Branch: BCA
Marks: 85

=== Example 2: Calculate Result ===

Total Marks: 253/300
Percentage: 84.33%

=== Example 3: Book Information ===

Book: Python Programming
Author: John Smith
Price: ₹499
Pages: 350

Book: Data Structures
Author: Jane Doe
Price: ₹599
Pages: 420

=== Example 4: Power Calculation ===

2^3 = 8
5^2 = 25
10^4 = 10000

=== Note: Order of arguments matters! ===

Correct order:
Name: Alice
Roll Number: 101
Branch: CSE
Marks: 92

Wrong order (will give wrong output):
Name: 101
Roll Number: Alice
Branch: 92
Marks: CSE
```

```
D:\Work\VSCode\Python_Codes\src>python Q77_V
Q77_VariableLengthParameters.py
=== Example 1: Sum with *args ===

Sum of 2 numbers: 30
Sum of 4 numbers: 100
Sum of 6 numbers: 105

=== Example 2: Student Marks ===

Student: Kaivalaya
Marks: (85, 90, 78, 92, 88)
Total: 433
Average: 86.60

Student: Alice
Marks: (75, 80, 85)
Total: 240
Average: 80.00

=== Example 3: Display Info with **kwargs ==

Details:
name: Kaivalaya
roll: 205
branch: BCA
city: Delhi

Details:
product: Laptop
price: 50000
brand: Dell
warranty: 2 years

=== Example 4: Complete Info (all types) ===

Name: Kaivalaya
Subjects: ('Python', 'Java', 'C++')
Additional Details:
  roll: 205
  branch: BCA
  year: 2

=== Example 5: Multiply All ===

Multiply 2 numbers: 20
Multiply 4 numbers: 120
Multiply 6 numbers: 720

=== Example 6: List Unpacking ===

Sum of list: 150

Details:
name: Bob
age: 20
city: Mumbai
```

```
D:\Work\VSCode\Python_Codes\src>python
Q70_DecimalConversion.py
Enter a decimal number: 2323

========================================
Decimal Number: 2323
========================================
Binary: 100100010011
Octal: 4423
Hexadecimal: 913
========================================

Manual Binary Conversion: 100100010011
```

```
D:\Work\VSCode\Python_Codes\src>python Q78_RoleBasedAccessDe
Q78_RoleBasedAccessDecorators.py
==============================================
   ROLE-BASED ACCESS CONTROL SYSTEM
==============================================

Available Users:
  1. Username: admin, Password: admin123, Role: admin
  2. Username: manager, Password: manager123, Role: manager
  3. Username: user, Password: user123, Role: user

==============================================
1. Login
2. View Profile
3. View Dashboard
4. Manage Team (Manager only)
5. Delete User (Admin only)
6. Logout
7. Exit

Enter choice: 7

Exiting system. Goodbye!
```

```
D:\Work\VSCode\Python_C
Q79_ClassAndObject.py
Student 1:
Name: Kaivalaya
Roll: 205
Marks: 85

Student 2:
Name: Alice
Roll: 101
Marks: 90
```

```
D:\Work\VSCode\Python_Codes\src>py
Q80_AbstractionVsEncapsulation.py
=== Abstraction ===
Car started
Car stopped

=== Encapsulation ===
Deposited: 500
Balance: 1500
```

```
D:\Work\VSCode\Python_Codes\sr
Q81_Encapsulation.py
Name: Kaivalaya
Marks: 85
Name: Kaivalaya, Marks: 90
```

```
D:\Work\VSCode\Python_Codes\src>p
Q82_Inheritance.py
Generic Animal makes a sound
Tommy barks: Woof Woof!
Kitty meows: Meow Meow!
```

```
D:\Work\VSCode\Python
Q83_Polymorphism.py
Circle area: 78.5
Rectangle area: 24
```

```
D:\Work\VSCode\Python_Codes\src>python Q84_LibraryManagementSystem.py
Q84_LibraryManagementSystem.py
Book 'Python Programming' added successfully
Book 'Data Structures' added successfully
Book 'Web Development' added successfully

=== Library Books ===
ID: 1, Title: Python Programming, Author: John Smith, Status: Available
ID: 2, Title: Data Structures, Author: Jane Doe, Status: Available
ID: 3, Title: Web Development, Author: Bob Wilson, Status: Available

Book 'Python Programming' issued successfully

=== Library Books ===
ID: 1, Title: Python Programming, Author: John Smith, Status: Issued
ID: 2, Title: Data Structures, Author: Jane Doe, Status: Available
ID: 3, Title: Web Development, Author: Bob Wilson, Status: Available

Book 'Python Programming' returned successfully

=== Library Books ===
ID: 1, Title: Python Programming, Author: John Smith, Status: Available
ID: 2, Title: Data Structures, Author: Jane Doe, Status: Available
ID: 3, Title: Web Development, Author: Bob Wilson, Status: Available
```

```
D:\Work\VSCode\Python_Codes\src>python
Q85_ShallowVsDeepCopy.py
Original: [1, 2, [3, 4], 5]
Shallow copy: [1, 2, [3, 4], 5]
Deep copy: [1, 2, [3, 4], 5]

After modifying original[2][0] to 999:
Original: [1, 2, [999, 4], 5]
Shallow copy: [1, 2, [999, 4], 5]
Deep copy: [1, 2, [3, 4], 5]
```

```
D:\Work\VSCode\Python_Codes\src>pyth
Q86_CheckCollinearity.py
Enter coordinates of three points:
x1: 34
y1: 2
x2: 5
y2: 34
x3: 23
y3: 54

The points are NOT collinear
```

```
D:\Work\VSCode\Python_Codes\src>python Q87_Ch
Q87_CheckQuadrant.py
Enter x coordinate: 23
Enter y coordinate: 23
Point (23.0, 23.0) lies in FIRST quadrant
```

```
D:\Work\VSCode\Python_Codes\src>pyt
Q88_TypeVsInstance.py
=== type() function ===
type(dog): <class '__main__.Dog'>
type(num): <class 'int'>
type(text): <class 'str'>

=== isinstance() function ===
isinstance(dog, Dog): True
isinstance(dog, Animal): True
isinstance(num, int): True
isinstance(text, str): True

=== Key Difference ===
type(dog) == Dog: True
type(dog) == Animal: False
isinstance(dog, Animal): True
```

```
D:\Work\VSCode\Python_Codes\src>python Q89_Create
Q89_CreateDestroyObjects.py
=== Creating Objects ===
Object created: Alice (Total objects: 1)
Object created: Bob (Total objects: 2)
Object created: Charlie (Total objects: 3)

Total objects: 3

=== Destroying Objects ===
Object destroyed: Alice (Remaining objects: 2)
Object destroyed: Bob (Remaining objects: 1)

Remaining objects: 1
Object destroyed: Charlie (Remaining objects: 0)

Final count: 0
```

```
D:\Work\VSCode\Python_Codes\src>
Q90_BasicTryExcept.py
Enter a number: 12345
Result: 0.0008100445524503848
```

```
D:\Work\VSCode\Python_Codes\src
Q91_TryExceptElse.py
Enter a number: 123456
Result: 8.100051840331778e-05
No error occurred!
```

```
D:\Work\VSCode\Python_Codes\sr
Q92_TryFinally.py
Enter a number: 234
Result: 0.042735042735042736
This always executes!
```

```
D:\Work\VSCode\Python_Codes\src>py
Q93_MultipleExceptions.py
Enter a number: 1234
Result: 0.008103727714748784
```

```
D:\Work\VSCode\Python_Codes\sr
Q94_RaisingExceptions.py
Enter your age: 20
Age 20 is valid
```

```
D:\Work\VSCode\Python_Codes\sr
Q95_UserDefinedExceptions.py
Enter marks: 100
Marks 100 is valid
```

```
D:\Work\VSCode\Python_Codes\src>pythor
Q96_TryExceptElseFinally.py
Enter a number: 1234
Result: 0.008103727714748784
Operation successful!
Execution completed!
```

```
D:\Work\VSCode\Python_Codes\src>python
Q97_FileHandlingException.py
File not found! Creating new file...
File created successfully!
File operation completed!
```

```
D:\Work\VSCode\Python_Codes\src>p
Q98_NestedTryExcept.py
Enter a number: 1234
Result: 0.008103727714748784
Index out of range!
```

```
D:\Work\VSCode\Python_Codes\src>python Q9
Q99_MultipleExceptionsOneLine.py
Enter a number: 1234
Error occurred: list index out of range
Error type: IndexError
```

```
D:\Work\VSCode\Python_Codes\src>py
Q100_RaiseReRaiseExceptions.py
Enter first number: 1234
Enter second number: 12345
Result: 0.09995949777237748
```

```
D:\Work\VSCode\Python_Codes\src>python
Q101_ExceptionsListOperations.py
Enter index: 3
Element at index 3: 40
Enter value to remove: 2
Error: list.remove(x): x not in list
```

```
D:\Work\VSCode\Python_Codes\src>python Q102_Exceptio
Q102_ExceptionsDictionaryLookup.py
Enter student name: Kevin
Error: Student 'Kevin' not found in database!
Available students: ['Alice', 'Bob', 'Charlie']
```