**Module 13. Recursion**

**TRUE/FALSE**

1. A function can call itself directly or indirectly through another function.

Answer: T

1. All recursive functions eventually lead to infinite recursion.

Answer: F

1. A recursive function must always have a base case to prevent infinite recursion.

Answer: T

1. Recursion is generally more memory-efficient than iteration.

Answer: F

1. In a recursive function, the base case is the condition that stops the recursion.

Answer: T

1. Recursive solutions are always faster than iterative solutions.

Answer: F

1. The factorial of a number can be computed using a recursive function.

Answer: T

1. The Fibonacci sequence can be generated using both recursion and iteration.

Answer: T

1. A recursive function without a base case will eventually cause a stack overflow error.

Answer: T

1. Recursive functions cannot return values.

Answer: F

1. A recursive function must always call itself with a smaller or simpler argument.

Answer: T

1. Recursion is only used for mathematical problems.

Answer: F

1. The GCD (Greatest Common Divisor) of two numbers can be found using recursion.

Answer: T

1. The efficiency of a recursive solution is not affected by the depth of recursion.

Answer: F

1. A recursive function can have multiple base cases.

Answer: T

**MULTIPLE CHOICE**

1. What is the primary reason to use recursion in programming?

a. To make the program run faster

b. To solve problems that can be broken down into smaller, similar problems

c. To use more memory

d. To make the code harder to understand

Answer: B

1. What happens if a recursive function does not have a base case?

a. The function will run indefinitely

b. The function will return an error immediately

c. The function will exit after a few iterations

d. The function will run faster

Answer: A

1. In the context of recursion, what is a base case?

a. The initial value of the function

b. The simplest instance of the problem that can be solved directly

c. The first recursive call

d. The last value of the function

Answer: B

1. Which of the following problems is NOT typically solved using recursion?

a. Factorial calculation

b. Sorting a list

c. Finding the maximum value in a list

d. Calculating the sum of an arithmetic series

Answer: D

1. Which of the following is a characteristic of a recursive function?

a. It always runs faster than iterative solutions

b. It includes a condition to stop the recursion

c. It never calls itself

d. It requires a for loop

Answer: B

1. In a recursive solution to find the sum of a list of numbers, what would the base case be?

a. When the list has one element

b. When the list is empty

c. When the sum is greater than zero

d. When the sum is less than zero

Answer: B

1. Which of the following best describes the role of the recursive case in a recursive function?

a. It sets the initial conditions for the recursion

b. It reduces the problem to a smaller instance

c. It stops the recursion

d. It calculates the final result

Answer: B

1. What is the base case for the recursive function to calculate the factorial of a number?

a. n = 2

b. n = 3

c. n = 0

d. n = -1

Answer: C

1. Which of the following problems is typically solved using recursion?

a. Calculating compound interest

b. Finding the nth Fibonacci number

c. Converting temperature from Celsius to Fahrenheit

d. Determining the area of a circle

Answer: B

1. Which statement is true about recursive functions?

a. They always lead to stack overflow errors

b. They must call themselves with a reduced problem size

c. They cannot return values

d. They are always more efficient than iterative functions

Answer: B

1. In a recursive solution for finding the greatest common divisor (GCD) of two numbers, what is the base case?

a. When the first number is zero

b. When the second number is zero

c. When both numbers are equal

d. When both numbers are prime

Answer: B

1. How does recursion help in solving problems related to tree traversal?

a. By using more memory

b. By avoiding the need for a stack

c. By simplifying the problem into smaller sub-problems

d. By speeding up the traversal

Answer: C

1. Which of the following is an example of a problem that is naturally suited for recursion?

a. Calculating simple interest

b. Performing a linear search in an array

c. Solving the Towers of Hanoi puzzle

d. Finding the square root of a number

Answer: C

1. In the context of recursion, what does the term "stack overflow" refer to?

a. Running out of memory due to deep recursion

b. An error in stack data structure implementation

c. Overflow of numerical values in the stack

d. Exceeding the limit of iterations in a loop

Answer: A

1. What is the role of the main function in a recursive program?

a. To handle the base case

b. To handle the recursive case

c. To initialize and start the recursive process

d. To terminate the recursive process

Answer: C

1. Which part of a recursive function ensures that the problem is eventually simplified to the base case?

a. The initialization step

b. The recursive call

c. The loop iteration

d. The return statement

Answer: B

1. What is the primary drawback of using recursion?

a. It makes the code harder to read

b. It is less intuitive than iteration

c. It can lead to stack overflow errors

d. It cannot be used for mathematical problems

Answer: C

1. Which of the following is true about tail recursion?

a. It occurs when the recursive call is the last operation in the function

b. It leads to infinite recursion

c. It cannot be optimized by the compiler

d. It always uses more memory than non-tail recursion

Answer: A

1. What is the main advantage of using recursion over iteration in some problems?

a. Recursion always uses less memory

b. Recursion can be more intuitive and easier to implement

c. Recursion is always faster

d. Recursion avoids the use of additional functions

Answer: B

1. In a recursive function to calculate the sum of elements in an array, what would be the recursive case?

a. Returning 0 if the array is empty

b. Returning the first element plus the sum of the remaining elements

c. Returning the last element of the array

d. Returning the sum of all elements directly

Answer: B