1. Which of the following is an example of direct recursion?
   1. **Factorial calculation**
   2. Binary search
   3. Fibonacci series generation
   4. Quick sort
2. Which of the following is an example of indirect recursion?
   1. Factorial calculation
   2. Binary search
   3. **Fibonacci series generation**
   4. Quick sort
3. What happens when a recursive function exceeds the maximum allowed recursion depth?
   1. **Stack overflow**
   2. Memory leak
   3. Buffer overflow
   4. Null pointer exception
4. In C++, the maximum recursion depth is limited by:
   1. The available memory
   2. **The size of the stack**
   3. The processor speed
   4. The compiler version
5. Which of the following is a potential issue with recursive functions?
   1. Excessive memory usage
   2. Infinite recursion
   3. Difficulty in understanding the code
   4. **All of the above**
6. What is tail recursion?
   1. Recursion involving a base case
   2. **Recursion involving a single recursive call at the end of the function**
   3. Recursion involving multiple recursive calls
   4. Recursion involving a recursive call within a loop
7. Tail recursion can be easily optimized by compilers using:
   1. Loop unrolling
   2. Function inlining
   3. **Tail call optimization**
   4. Code elimination
8. Which of the following recursion types is generally more memory-efficient?
   1. **Tail recursion**
   2. Non-tail recursion
   3. Direct recursion
   4. Indirect recursion
9. A recursive function is said to be non-tail recursive if:
   1. It calls another function
   2. It has multiple base cases
   3. **It has multiple recursive calls**
   4. It performs a calculation at each recursive step
10. Which of the following recursion types typically results in a larger memory footprint?
    1. Tail recursion
    2. **Non-tail recursion**
    3. Direct recursion
    4. Indirect recursion
11. In tail recursion, the recursive call is:
    1. Executed before any other statements in the function
    2. **Executed after all other statements in the function**
    3. Executed in parallel with other statements in the function
    4. Executed based on a conditional statement
12. What is the advantage of using tail recursion?
    1. **It reduces memory usage**
    2. It improves code readability
    3. It eliminates the need for a base case
    4. It increases the execution speed
13. Which of the following statements is true regarding indirect recursion?
    1. It involves a function calling itself indirectly through another function
    2. It involves a function calling itself directly
    3. **It involves two or more functions calling each other in a cycle**
    4. It involves a function calling multiple other functions
14. When using indirect recursion, care should be taken to:
    1. Avoid infinite recursion
    2. Avoid stack overflow
    3. Ensure termination conditions are met
    4. **All of the above**
15. Which of the following is an example of direct recursion?
    1. **Tower of Hanoi problem**
    2. Euclidean algorithm for finding the greatest common divisor
    3. Depth-first search algorithm
    4. Merge sort
16. Which of the following is an example of indirect recursion?
    1. Tower of Hanoi problem
    2. **Euclidean algorithm for finding the greatest common divisor**
    3. Depth-first search algorithm
    4. Merge sort
17. When a recursive function is called, which of the following is stored in the function call stack?
    1. Return address
    2. Local variables
    3. Function parameters
    4. **All of the above**
18. In tail recursion, the recursive call is performed:
    1. Before the current function call completes
    2. **After the current function call completes**
    3. In parallel with the current function call
    4. At any arbitrary point during the function call
19. Which of the following is true regarding stack overflow?
    1. **It occurs when the stack exceeds its maximum size**
    2. It occurs when a recursive function has too many base cases
    3. It occurs when a function has too many local variables
    4. It occurs when a function returns too many values
20. In C++, which data structure is commonly used for implementing the function call stack?
    1. Array
    2. Linked list
    3. Queue
    4. **Stack**
21. Non-tail recursion can be transformed into tail recursion by using:
    1. Dynamic programming
    2. Memoization
    3. **Iteration**
    4. Stack manipulation
22. Which of the following statements about tail recursion is false?
    1. It is equivalent to an iterative solution
    2. It can always be optimized by compilers
    3. **It eliminates the need for a base case**
    4. It reduces memory usage
23. Which type of recursion is generally harder to analyze and understand?
    1. Direct recursion
    2. **Indirect recursion**
    3. Tail recursion
    4. Non-tail recursion
24. What happens if a non-tail recursive function is not properly designed?
    1. **Stack overflow may occur**
    2. The program may terminate early
    3. The function may not compile
    4. The program may run indefinitely
25. The maximum recursion depth in C++ can be increased by:
    1. **Using a larger stack size**
    2. Using compiler optimizations
    3. Decreasing the size of local variables
    4. None of the above
26. When using recursion, it is important to ensure:
    1. A proper termination condition exists
    2. The base case is handled correctly
    3. The recursive call converges towards the base case
    4. **All of the above**
27. Which of the following is an advantage of using recursion?
    1. **It can simplify the problem-solving process**
    2. It always results in faster code execution
    3. It eliminates the need for loops
    4. It reduces the size of the code
28. Which of the following is not an advantage of using tail recursion?
    1. It reduces memory usage
    2. **It eliminates the need for a base case**
    3. It can be easily optimized by compilers
    4. It simplifies the code structure
29. In indirect recursion, the functions involved in the cycle can be called in any order.
    1. **True**
    2. False
30. Tail recursion can be converted into an equivalent iterative solution.
    1. **True**
    2. False
31. Non-tail recursion is always less memory-efficient than tail recursion.
    1. **True**
    2. False
32. A function that does not have a termination condition will result in an infinite recursion.
    1. **True**
    2. False
33. Recursion is always a more efficient solution compared to iteration.
    1. True
    2. **False**