Q1–Q20: Multiple Choice Questions (MCQs)

Q1. What will `recur(4)` return in the following code?

```
int recur(int n) {
    if (n == 0) return 0;
    return n + recur(n - 1);
}
```

A) 4
B) 6
C) 10
D) 16

Q2. What is the output of `f(3)`?

```
int f(int n) {
    if (n <= 1) return n;
    return f(n - 1) + f(n - 2);
}
```

A) 2
B) 3
C) 4
D) 5

Q3. Which of the following recursive function definitions calculates factorial correctly?

```
int fact(int n) {
    if (n <= 1) return 1;
    return n * fact(n - 1);
}
```

A) Correct
B) Incorrect
C) Depends on base case
D) None

Q4. What does the following function compute?

```
int recur(int n) {
    if (n <= 1) return 1;
    return recur(n - 1) + recur(n - 1);
}
```

A) Factorial
B) Power of 2
C) Fibonacci
D) Square of n

Q5. What is the space complexity of the below recursion?

```
int sum(int n) {
    if (n == 0) return 0;
    return n + sum(n - 1);
}
```

A) O(1)
B) O(n)
C) O(log n)
D) $O(n^2)$

Q6. What happens if there is no base case in recursion?

A) The function will return 0
B) Infinite recursion and StackOverflow
C) Nothing, recursion ends automatically
D) Syntax error

Q7. What will be the output for `fun(4)`?

```
int fun(int n) {
    if (n == 1) return 1;
    return fun(n - 1) * n;
}
```

A) 4
B) 24
C) 10
D) 5

Q8. The recurrence relation `T(n) = 2*T(n/2) + n` corresponds to which algorithm's time complexity?

A) Bubble Sort
B) Merge Sort
C) Insertion Sort
D) Linear Search

Q9. What is the return value of `func(3)`?

```
int func(int n) {
    if (n <= 1) return 1;
    return func(n - 1) + func(n - 1);
}
```

A) 3
B) 4
C) 8
D) 2

Q10. Which of the following recursion types is used in this function?

```
int f(int n) {
    if (n == 0) return 0;
    return f(n - 1) + n;
}
```

A) Tail recursion
B) Head recursion

C) Tree recursion
D) Linear recursion

Q11–Q20: Multiple Choice Questions (MCQs)

Q11. What does the following function do?

```
int f(int n) {
    if (n == 0) return 1;
    return f(n / 2);
}
```

A) Returns log base 2 of n
B) Infinite recursion
C) Always returns 1
D) Stack overflow for large n

Q12. What is the output of `foo(3)`?

```
int foo(int n) {
    if (n == 0) return 1;
    return n * foo(n - 1);
}
```

A) 6
B) 9
C) 3
D) 0

Q13. Which problem is best solved using divide and conquer recursion?

A) Linear Search
B) Bubble Sort
C) Merge Sort
D) Sequential Traversal

Q14. How many times is `fun()` called for `fun(3)` in this code?

```
int fun(int n) {
    if (n <= 1) return 1;
    return fun(n - 1) + fun(n - 1);
}
```

A) 3
B) 5
C) 7
D) 15

Q15. What is the return value of `fun(2)`?

```
int fun(int n) {
    if (n == 0) return 2;
    return 2 * fun(n - 1);
}
```

A) 2
B) 4

C) 8
D) 6

Q16. What will the following code print for `solve(3)`?

```
int solve(int n) {
    if (n == 0) return 0;
    return n + solve(n - 2);
}
```

A) 6
B) 4
C) 3
D) 2

Q17. Which case is a tail-recursive function?

```
int tail(int n, int acc) {
    if (n == 0) return acc;
    return tail(n - 1, acc + n);
}
```

A) Yes
B) No
C) Depends on input
D) Only if n > 0

Q18. Which of the following is not an advantage of recursion?

A) Simpler code
B) Less memory usage
C) Elegant divide-and-conquer solutions
D) Reduces code complexity for tree traversal

Q19. What is returned from `mystery(4)`?

```
int mystery(int n) {
    if (n == 0) return 1;
    return mystery(n - 1) * 2;
}
```

A) 4
B) 8
C) 16
D) 1

Q20. Which recursive call tree has the largest number of calls for `n = 5`?

A) Factorial
B) Fibonacci
C) Linear recursion
D) Tail recursion

Q21–Q40: Output-Based Recursion Questions

Q21. What is the output of the following?

```
void recur(int n) {
    if (n == 0) return;
```

```
    recur(n - 1);
    System.out.print(n + " ");
}
recur(3);
```

Q22. What is the output of this?

```
    void recur(int n) {
        if (n == 0) return;
        System.out.print(n + " ");
        recur(n - 1);
    }
    recur(3);
```

Q23. Predict the output:

```
    void recur(int n) {
        if (n <= 0) return;
        recur(n - 1);
        System.out.print(n);
        recur(n - 2);
    }
    recur(3);
```

Q24. What is the output?

```
    void recur(int n) {
        if (n <= 0) return;
        recur(n - 2);
        System.out.print(n + " ");
    }
    recur(5);
```

Q25. What is printed?

```
    void fun(int n) {
        if (n == 0) return;
        fun(n / 2);
        System.out.print(n % 2);
    }
    fun(5);
```

Q26. Output of this code?

```
    void print(int n) {
        if (n <= 0) return;
        System.out.print(n + " ");
        print(n - 1);
        System.out.print(n + " ");
    }
    print(3);
```

Q27. What is printed?

```
int fun(int n) {
    if (n < 2) return 1;
    return fun(n - 1) + fun(n - 2);
}
System.out.println(fun(5));
```

Q28. Predict the output

```
int count = 0;
int f(int n) {
    if (n == 0) return 1;
    count++;
    return f(n - 1);
}
f(5);
System.out.println(count);
```

Q29. Output of the below:

```
int sum = 0;
void recur(int n) {
    if (n == 0) return;
    sum += n;
    recur(n - 1);
}
recur(4);
System.out.println(sum);
```

Q30. Output of the following code:

```
void recur(int n) {
    if (n == 0) return;
    recur(n - 1);
    System.out.print(n * n + " ");
}
recur(3);
```

Q31. What is printed?

```
void printDigits(int n) {
    if (n == 0) return;
    printDigits(n / 10);
    System.out.print(n % 10 + " ");
}
printDigits(123);
```

Q32. Output of the function:

```
int mystery(int n) {
```

```
        if (n <= 1) return 1;
        return mystery(n - 1) + n;
    }
    System.out.println(mystery(5));
```

Q33. What will be printed?

```
    void mystery(int n) {
        if (n == 0) return;
        System.out.print(n + " ");
        mystery(n / 2);
    }
    mystery(8);
```

Q34. Output of the following:

```
    void printSeries(int n) {
        if (n <= 0) return;
        printSeries(n - 1);
        System.out.print((char)(n + 64) + " ");
    }
    printSeries(3);
```

Q35. What does this function print?

```
    void fun(int n) {
        if (n == 0) return;
        fun(n - 1);
        System.out.print(n + " ");
        fun(n - 1);
    }
    fun(2);
```

Q36. What is printed here?

```
    void recur(int n) {
        if (n == 0) return;
        System.out.print(n + " ");
        recur(n / 2);
    }
    recur(10);
```

Q37. Output of following recursive code:

```
    void recur(int a, int b) {
        if (a == 0 || b == 0) return;
        recur(a - 1, b - 1);
        System.out.print(a + "," + b + " ");
    }
    recur(3, 3);
```

Q38. Predict the output:

```
void solve(int n) {
    if (n < 0) return;
    System.out.print(n + " ");
    solve(n - 3);
}
solve(9);
```

Q39. Output of code below:

```
void oddEven(int n) {
    if (n == 0) return;
    if (n % 2 == 0) System.out.print("E ");
    else System.out.print("O ");
    oddEven(n - 1);
}
oddEven(5);
```

Q40. Output for this recursive call:

```
void doubleRecur(int n) {
    if (n == 0) return;
    System.out.print(n + " ");
    doubleRecur(n - 1);
    doubleRecur(n - 1);
}
doubleRecur(2);
```

Q41–Q50: Debugging-Based Recursion Questions

Each question contains a bug (logical or syntactical).You need to **identify and fix the issue**. The correct version is provided.

Q41. What's wrong with this factorial function?

```
int fact(int n) {
    if (n == 0) return 0;
    return n * fact(n - 1);
}
```

Q42. What is wrong here?

```
int fib(int n) {
    return fib(n - 1) + fib(n - 2);
}
```

Q43. The code causes a StackOverflow. Why?

```
int call(int n) {
    return call(n);
}
```

Q44. Find and fix the error.

```
int power(int base, int exp) {
    if (exp == 0) return 0;
    return base * power(base, exp - 1);
}
```

Q45. What's the problem here?

```
int sum(int n) {
    if (n == 1) return 1;
    return n + sum(n + 1);
}
```

Q46. Why is this function not terminating?

```
void print(int n) {
    System.out.println(n);
    print(n - 1);
}
```

Q47. Correct the logic to print digits of number `n` in reverse.

```
void reverseDigits(int n) {
    if (n == 0) return;
    reverseDigits(n / 10);
    System.out.print(n % 10);
}
```

Q48. Fix the bug in this countdown program.

```
void countdown(int n) {
    if (n == 0) System.out.println("Blast off!");
    countdown(n - 1);
}
```

Q49. The program is supposed to compute `a^b`, but it's incorrect. Why?

```
int pow(int a, int b) {
    if (b == 0) return 0;
    return a * pow(a, b - 1);
}
```

Q50. Fix the logic error in this string reversal function.

```
String reverse(String str) {
    if (str.length() == 0) return "";
    return str.charAt(0) +
reverse(str.substring(1));
}
```