Q36–Q50: Function Return and Nested Calls?

Q36. What is the output of the following?

Pseudocode Idea: Function returns value; result is used in next function

```
public class Main {
    public static int add(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {
        int x = add(add(2, 3), 4);
        System.out.println(x);
    }
}
```

Q37. Output of nested return call

```
public class Main {
    public static int square(int x) {
        return x * x;
    }

    public static void main(String[] args) {
        int res = square(square(2));
        System.out.println(res);
    }
}
```

Q38. Trace the output

```
public class Main {
    public static int foo(int x) {
        if (x == 0) return 1;
        return x * foo(x - 1);
    }

    public static void main(String[] args) {
        System.out.println(foo(4));
    }
}
```

Q39. Predict the result

```java
public class Main {
    public static int mul(int a, int b) {
        return a * b;
    }

    public static int add(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {
        int result = mul(add(1, 2), add(2, 3));
        System.out.println(result);
    }
}
```

Q40. What is printed?

```java
public class Main {
    public static int f(int x) {
        return x + 1;
    }

    public static void main(String[] args) {
        int a = f(f(f(1)));
        System.out.println(a);
    }
}
```

Q41. What is the output?

```java
public class Main {
    public static int f(int a) {
        if (a == 1) return 1;
        return a + f(a - 1);
    }

    public static void main(String[] args) {
        int x = f(3);
        System.out.println(x);
    }
}
```

Q42. Recursive return-based trace

```
public class Main {
    public static int mystery(int x) {
        if (x == 0) return 1;
        return 2 * mystery(x - 1);
    }

    public static void main(String[] args) {
        System.out.println(mystery(4));
    }
}
```

Q43. What will be returned?

```
public class Main {
    public static int process(int x) {
        return x * 2;
    }

    public static void main(String[] args) {
        int x = 3;
        int y = process(process(x));
        System.out.println(y);
    }
}
```

Q44. Output?

```
public class Main {
    public static int call(int x) {
        return x + 2;
    }

    public static void main(String[] args) {
        int result = call(call(call(0)));
        System.out.println(result);
    }
}
```

Q45. What is printed?

```
public class Main {
    public static int calc(int x) {
        return x * x + 1;
    }

    public static void main(String[] args) {
        System.out.println(calc(2) + calc(3));
    }
}
```

Q46. Predict the output:

```
public class Main {
    public static int f(int x) {
        return x % 2 == 0 ? x : f(x - 1);
    }

    public static void main(String[] args) {
        System.out.println(f(7));
    }
}
```

Q47. Output of chaining calls:

```
public class Main {
    public static int f(int a) {
        return a + 1;
    }

    public static int g(int a) {
        return f(a) * 2;
    }

    public static void main(String[] args) {
        System.out.println(g(3));
    }
}
```

Q48. Return logic test

```
public class Main {
    public static int decide(int a) {
        if (a > 5) return 10;
        return 5;
    }

    public static void main(String[] args) {
        System.out.println(decide(3) + decide(6));
    }
}
```

Q49. Function calling itself indirectly

```
public class Main {
    public static int funA(int n) {
        if (n <= 0) return 0;
        return n + funB(n - 1);
    }

    public static int funB(int n) {
        if (n <= 0) return 0;
        return n + funA(n / 2);
    }

    public static void main(String[] args) {
        System.out.println(funA(4));
    }
}
```

Q50. Function returning boolean logic

```
public class Main {
    public static boolean isEven(int x) {
        return x % 2 == 0;
    }

    public static void main(String[] args) {
        System.out.println(isEven(6));
    }
}
```