JAVA TEST 1

Section 1: MCQs & Output-Based Questions (10 Questions)

1. What will be the output of the following Java code?

```java
public class Test {
    public static void main(String[] args) {
        int a = 5;
        int b = 10;
        while (a-- > 0 && ++b < 20) {
            System.out.println(a + " " + b);
        }
    }
}
```

a) Infinite loop
b) Compilation error
c) Some output and then stops
d) No output

2. What is the output of the following program?

```java
public class Test {
    public static void main(String[] args) {
        int x = 10;
        System.out.println(x++ + x++ + x++);
    }
}
```

a) 10 11 12
b) 33
c) 36
d) Compilation error

3. Consider the following function:

```java
public static void tricky(int n) {
    if (n > 0) {
        tricky(n - 1);
        System.out.print(n + " ");
        tricky(n - 2);
    }
}
public static void main(String[] args) {
    tricky(3);
}
```

What will be the output?
a) 3 2 1
b) 1 2 3
c) 1 2 3 1
d) 1 3 2 1

4. How many times will `"Hello"` be printed in this loop?

```
public class Test {
    public static void main(String[] args) {
        int i = 1, j = 1;
        while (++i < 5 || j++ < 3) {
            System.out.println("Hello");
        }
    }
}
```

a) 5
b) 6
c) 3
d) Infinite

5. What is the output of this function?

```
static int func(int n) {
    if (n == 0) return 0;
    return func(n / 2) + n % 2;
}
public static void main(String[] args) {
    System.out.println(func(11));
}
```

a) 5
b) 6
c) 3
d) 4

6. What does this recursive function do?

```
static int mystery(int n) {
    if (n < 10) return n;
    return (n % 10) + mystery(n / 10);
}
public static void main(String[] args) {
    System.out.println(mystery(253));
}
```

a) Counts the digits of the number
b) Finds the sum of the digits
c) Finds the factorial
d) Prints digits in reverse

7. Identify the issue in the following function:

```
public static int factorial(int n) {
    if (n == 1) return 1;
    return n * factorial(n);
}
```

a) No issue, it works fine
b) Infinite recursion (StackOverflowError)
c) Compilation error
d) Wrong output

8. What is the output of this Java program?

```
public class Test {
    public static void main(String[] args) {
        for (int i = 0; i < 3; i++) {
            for (int j = i; j < 3; j++) {
                System.out.print(i + "" + j + " ");
            }
        }
    }
}
```

a) 00 01 02 11 12 22
b) 00 01 02 11 12 21
c) 00 01 02 11 12 13
d) 00 01 02 10 11 12

9. What will be the output of this Java code?

```
public class Test {
    public static void main(String[] args) {
        for (int i = 1; i < 10; i *= 2) {
            System.out.println(i);
        }
    }
}
```

a) 1 2 3 4 5 6 7 8 9
b) 1 2 4 8
c) 1 2 4
d) Compilation error

10. What will be printed by this function call?

```
static void strange(int n) {
    if (n == 0) return;
    System.out.print(n + " ");
    strange(n / 2);
}
public static void main(String[] args) {
    strange(10);
}
```

a) 10 5 2 1
b) 10 5 2 1 0
c) 10 9 8 7 6 5 4 3 2 1
d) Compilation error

Section 2: (10 Questions)

11. Write a function to find the sum of digits of a number using recursion but without using the modulus (%) operator.

12. Write a Java function that reverses a number without converting it to a string or using arrays.

13. Write a function that prints all numbers from `n` to `1` and then back to `n` using recursion.

Example:
Input: `n = 3`
Output: `3 2 1 1 2 3`

14. Implement a function that prints the following pattern:

```
1 2 3 4 5 6 7
 2 3 4 5 6 7
  3 4 5 6 7
   4 5 6 7
    5 6 7
     6 7
      7
     6 7
    5 6 7
   4 5 6 7
  3 4 5 6 7
 2 3 4 5 6 7
1 2 3 4 5 6 7
```

15. Write a function to check whether a number is Armstrong or not using recursion.

Example:
Input: `153`
Output: `True`

16. Write a function that finds all prime numbers in a given range `[L, R]` using recursion.

17. Write a function that calculates `n^p` (n raised to the power p) using recursion without using the `Math.pow()` method.

18. Given a number n, print its binary representation without using recursion.

Example:
Input: `5`
Output: `101`

19. Write a recursive function that prints a given number in reverse order.

Example:
Input: `1234`
Output: `4321`

20. Write a function that swaps the first and last digit of a given number without using arrays or strings.

Example:
Input: 12345
Output: 52341