Array Tracing Problems

Problem 1

```
public static int mystery1(int[] arr) {
    int result = 0;
    for (int i = 0; i < arr.length; i++) {
        result = result + arr[i];
    }
    return result;
}
```

What does mystery1(new int[]{2, 5, 3, 8}) return?

Problem 2

```
public static void mystery2(int[] arr) {
    for (int i = 0; i < arr.length; i += 2) {
        System.out.print(arr[i] + " ");
    }
}
```

What does mystery2(new int[]{10, 20, 30, 40, 50}) print?

Problem 3

```
public static int mystery3(int[] arr) {
    int count = 0;
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] > 5) {
            count++;
        }
    }
    return count;
}
```

What does mystery3(new int[]{3, 7, 2, 9, 5, 8}) return?

Problem 4

```
public static void mystery4(int[] arr) {
    for (int i = arr.length - 1; i >= 0; i--) {
        System.out.print(arr[i] + " ");
    }
}
```

What does mystery4(new int[]{1, 2, 3, 4}) print?

Problem 5: String Accumulation

```
public static String mystery5(String[] arr) {
    String result = "";
```

```
    for (int i = 0; i < arr.length; i++) {
        result = result + arr[i];
    }
    return result;
}
```

What does mystery5(new String[]{“cat”, “dog”, “pig”}) return?

Problem 6

```
public static int mystery6(int[] arr) {
    int max = arr[0];
    for (int i = 1; i < arr.length; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }
    return max;
}
```

What does mystery6(new int[]{4, 9, 2, 11, 5}) return?

Problem 7

```
public static void mystery7(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] % 2 == 0) {
            arr[i] = arr[i] * 2;
        }
    }
    System.out.print(arr[0] + " " + arr[1] + " " + arr[2]);
}
```

What does mystery7(new int[]{3, 4, 5}) print?

Problem 8

```
public static int mystery8(int[] arr) {
    int result = 1;
    for (int value : arr) {
        result = result * value;
    }
    return result;
}
```

What does mystery8(new int[]{2, 3, 4}) return?

Problem 9

```
public static boolean mystery9(int[] arr) {
    for (int i = 0; i < arr.length - 1; i++) {
        if (arr[i] > arr[i + 1]) {
```

```
            return false;
        }
    }
    return true;
}
```

What does mystery9(new int[]{1, 3, 5, 4, 7}) return?

Problem 10

```
public static int mystery10(int[] arr) {
    int result = 0;
    for (int i = 0; i < arr.length; i++) {
        if (i % 2 == 0) {
            result = result + arr[i];
        } else {
            result = result - arr[i];
        }
    }
    return result;
}
```

What does mystery10(new int[]{10, 3, 8, 5, 6}) return?

Answer Key

1. **16** (sum of all elements: 2+5+3+8)
2. **10 30 50** (prints elements at indices 0, 2, 4)
3. **3** (counts elements greater than 5: 7, 9, 8)
4. **4 3 2 1** (prints array backwards)
5. **"catdogpig"** (concatenates all strings)
6. **11** (finds the maximum value)
7. **3 8 5** (doubles even numbers: 3 stays 3, 4 becomes 8, 5 stays 5)
8. **24** (multiplies all elements: 2×3×4)
9. **false** (checks if sorted in ascending order; fails at 5>4)
10. **16** (alternates add/subtract: 10-3+8-5+6)

Would you like me to create more problems, adjust the difficulty level, or focus on specific array concepts?