**1. What is the default value assigned to array elements in C#?**

- **Default:** 0 for numeric types, false for bool, null for reference types.

---

**2. Difference between Array.Clone() and Array.Copy()**

- Clone() → Returns a **new array** with a shallow copy of elements, same length.
- Copy() → Copies elements from one array to another **existing** array; can specify range.

---

**3. Difference between GetLength() and Length for multi-dimensional arrays**

- Length → Total number of elements in all dimensions.
- GetLength(dimension) → Number of elements in a **specific dimension**.

---

**4. Difference between Array.Copy() and Array.ConstrainedCopy()**

- Copy() → Copies without guaranteeing rollback on failure.
- ConstrainedCopy() → Ensures **all-or-nothing** copy; if it fails, no partial changes occur.

---

**5. Why is foreach preferred for read-only operations on arrays?**

- Prevents accidental modification, more readable, no index management required.

---

**6. Why is input validation important when working with user inputs?**

- Prevents errors, crashes, and security issues by ensuring data is correct before processing.

---

**7. How can you format the output of a 2D array for better readability?**

- Use **tabs (\t)**, **alignment methods (PadLeft/PadRight)**, or String.Format() to align columns.

**8. When should you prefer a switch statement over if-else?**

- When checking **one variable** against multiple constant values for cleaner, faster code.

---

**9. What is the time complexity of Array.Sort()?**

- **O(n log n)** (uses QuickSort or IntroSort internally depending on the type).

---

**10. Which loop (for or foreach) is more efficient for calculating the sum of an array, and why?**

- **for loop** is slightly more efficient since foreach adds an enumerator overhead, but the difference is negligible for small arrays.