**1. Insert an element at a specific position**

**Q:** How do you insert an element at a given index in an array?  
**A:**  
Since arrays have fixed size in Java, we must create a new array or shift elements to make space.

int[] arr = {10, 20, 30, 40};

int pos = 2, val = 25;

for (int i = arr.length - 1; i > pos; i--) {

arr[i] = arr[i - 1];

}

arr[pos] = val;

**2. Delete an element at a specific position**

**Q:** How do you delete an element at index pos?  
**A:**  
Shift all elements after pos one position to the left.

int[] arr = {10, 20, 30, 40};

int pos = 1;

for (int i = pos; i < arr.length - 1; i++) {

arr[i] = arr[i + 1];

}

**3. Linear Search**

**Q:** What is linear search and when is it used?  
**A:**  
Linear search checks each element until the target is found or the array ends.

int[] arr = {5, 8, 3, 9};

int target = 3;

boolean found = false;

for (int i = 0; i < arr.length; i++) {

if (arr[i] == target) { found = true; break; }

}

**4. Left Shift**

**Q:** What does a left shift do in arrays?  
**A:**  
Removes the first element and shifts all others left, last becomes 0 or unused.

int[] arr = {1, 2, 3, 4};

for (int i = 0; i < arr.length - 1; i++) {

arr[i] = arr[i + 1];

}

arr[arr.length - 1] = 0;

**5. Right Shift**

**Q:** How is right shift different from left shift?  
**A:**  
Right shift moves all elements to the right, last becomes empty/0.

int[] arr = {1, 2, 3, 4};

for (int i = arr.length - 1; i > 0; i--) {

arr[i] = arr[i - 1];

}

arr[0] = 0;

**6. Left Rotation**

**Q:** What is left rotation?  
**A:**  
Moves first element to last position while shifting others left.

int[] arr = {1, 2, 3, 4};

int first = arr[0];

for (int i = 0; i < arr.length - 1; i++) {

arr[i] = arr[i + 1];

}

arr[arr.length - 1] = first;

**7. Right Rotation**

**Q:** What is right rotation?  
**A:**  
Moves last element to first position while shifting others right.

int[] arr = {1, 2, 3, 4};

int last = arr[arr.length - 1];

for (int i = arr.length - 1; i > 0; i--) {

arr[i] = arr[i - 1];

}

arr[0] = last;

**8. Reverse an Array**

**Q:** How do you reverse an array in-place?  
**A:**  
Swap first with last, second with second last, etc.

int[] arr = {1, 2, 3, 4};

for (int i = 0, j = arr.length - 1; i < j; i++, j--) {

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

**📍 Phase 2 – Java Built-in Tools**

*(Arrays class methods: sort, binarySearch, copy, equals)*

**9. Arrays.toString()**

**Q:** Why use Arrays.toString()?  
**A:**  
It prints arrays in readable form.

import java.util.Arrays;

int[] arr = {1, 2, 3};

System.out.println(Arrays.toString(arr)); // [1, 2, 3]

**10. Arrays.sort()**

**Q:** How does Arrays.sort() work?  
**A:**  
Sorts elements in ascending order (dual-pivot quicksort for primitives).

int[] arr = {3, 1, 2};

Arrays.sort(arr); // [1, 2, 3]

**11. Arrays.binarySearch()**

**Q:** What does Arrays.binarySearch() require?  
**A:**  
Array must be sorted; returns index if found, else negative position.

int[] arr = {1, 3, 5};

System.out.println(Arrays.binarySearch(arr, 3)); // 1

**12. Arrays.copyOf()**

**Q:** How to copy first N elements?  
**A:**

int[] arr = {1, 2, 3, 4};

int[] copy = Arrays.copyOf(arr, 2); // [1, 2]

**13. Arrays.equals()**

**Q:** How to check if two arrays are same?  
**A:**

int[] a = {1, 2};

int[] b = {1, 2};

System.out.println(Arrays.equals(a, b)); // true

**14. Arrays.copyOfRange()**

**Q:** How to copy a part of an array?

int[] arr = {1, 2, 3, 4};

int[] part = Arrays.copyOfRange(arr, 1, 3); // [2, 3]

**📍 Phase 3 – Problem Solving**

**15. Remove Duplicates from Sorted Array**

int[] arr = {1, 1, 2};

int idx = 1;

for (int i = 1; i < arr.length; i++) {

if (arr[i] != arr[i - 1]) arr[idx++] = arr[i];

}

**16. Merge Two Sorted Arrays**

int[] a = {1,3,5}, b = {2,4,6};

int[] res = new int[a.length + b.length];

int i=0,j=0,k=0;

while(i<a.length && j<b.length) res[k++] = (a[i]<b[j]) ? a[i++] : b[j++];

**17. Count Occurrences**

int[] arr = {1,2,2,3};

int target = 2, count=0;

for(int x: arr) if(x==target) count++;

**18. Check Palindrome Array**

int[] arr = {1,2,2,1};

boolean isPal = true;

for(int i=0,j=arr.length-1;i<j;i++,j--)

if(arr[i]!=arr[j]) isPal=false;

**19. Move Zeroes to End**

int[] arr = {0,1,0,3,12};

int idx=0;

for(int x: arr) if(x!=0) arr[idx++]=x;

while(idx<arr.length) arr[idx++]=0;

**20. Bubble Sort (No built-in)**

int[] arr = {3,2,1};

for(int i=0;i<arr.length-1;i++)

for(int j=0;j<arr.length-i-1;j++)

if(arr[j]>arr[j+1]){

int t=arr[j]; arr[j]=arr[j+1]; arr[j+1]=t;

}