

EXPERIMENT FOR PROGRAMMING ABILITY AND LOGIC BUILDING-1 (JAVA)

Pdf-2

EXPERIMENT-1

A screenshot of a Java code editor on LeetCode. The code is a solution for the problem "k-th Smallest Element in an Unsorted Array". It sorts the array and returns the element at index $k-1$. The editor shows the code in a Java (21) tab, with a Start Timer button. The output window shows "Problem Solved Successfully". Test cases passed: 1121 / 1121. Attempts: Correct / Total: 4 / 4. Accuracy: 100%. Time taken: 0.78.

```
1 class Solution {
2     public int kthSmallest(int[] arr, int k) {
3         // Code here
4         Arrays.sort(arr);
5         return arr [k-1];
6     }
7 }
```

EXPERIMENT-4

A screenshot of a LeetCode submission for the problem "Find the Duplicate Number". The code is a solution for the problem. It uses a sorting approach to find the duplicate number. The submission was accepted with 59/59 testcases passed by ShrutiSingh_07 at Feb 11, 2026 09:46. The runtime is 39 ms (Beats 6.52%) and memory usage is 79.36 MB (Beats 91.60%). The code editor shows the Java code and the test result section indicates it is Accepted with a runtime of 1 ms.

```
1 class Solution {
2     public int findDuplicate(int[] nums) {
3         Arrays.sort(nums);
4
5         for(int i = 0; i < nums.length - 1; i++) {
6             if(nums[i] == nums[i + 1]) {
7                 return nums[i];
8             }
9         }
10    }
11 }
```

EXPERIMENT-5

The screenshot shows a browser window with three tabs: "JAVA EXP-2 - Google Docs", "Find the Duplicate Number - Le", and "Merge Without Extra Space | Pr". The main content is the GeeksforGeeks problem page for "Merge Two Sorted Arrays".

Output Window:

- Compilation Results:** Custom Input, Y.O.G.I. (AI Bot)
- Problem Solved Successfully** (with a green checkmark)
- Test Cases Passed:** 1111 / 1111
- Attempts : Correct / Total:** 1 / 1
- Accuracy:** 100%
- Points Scored:** 4 / 4
- Time Taken:** 1.05
- Your Total Score:** 19 ↑

Code Editor (Java 21):

```
1 class Solution {
2     public void mergeArrays(int a[], int b[]) {
3         // code here
4
5         int n = a.length;
6         int m = b.length;
7         for(int i = 0; i < n; i++) {
8             if(a[i] > b[0]) {
9
10                 int temp = a[i];
11                 a[i] = b[0];
12                 b[0] = temp;
13
14                 int first = b[0];
15                 int k;
16
17                 for(k = 1; k < m && b[k] < first; k++) {
18                     b[k-1] = b[k];
19                 }
20                 b[k-1] = first;
21             }
22         }
23     }
24 }
25 }
```

Buttons: Start Timer, Custom Input, Compile & Run, Submit, Ctrl + Enter.

EXPERIMENT-6