

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

pdf-1

EXPERIMENT-1

The screenshot shows the 'Reverse an Array' problem page on GeeksforGeeks. The 'Output Window' on the left indicates the problem was solved successfully with 1115 test cases passed out of 1115, 1 attempt correct out of 1, 100% accuracy, 2 points scored out of 2, and a time taken of 0.8 seconds. The 'Compilation Results' tab is active. The code editor on the right shows a Java solution for reversing an array using a two-pointer approach. The code is as follows:

```
1 class Solution {
2     public void reverseArray(int arr[]) {
3         // code here
4         int i=0;
5         int j=arr.length-1;
6         while(i<j){
7             int temp=arr[i];
8             arr[i]=arr[j];
9             arr[j]=temp;
10            i++;
11            j--;
12        }
13    }
14    public static void main(String[] args){
15        int[] arr={1,4,3,2,6,5};
16        System.out.println("Reversed Array: "+ Arrays.toString(arr));
17    }
18 }
```

EXPERIMENT-2

The screenshot shows the 'Find Minimum and Maximum Element in an Array' problem page on GeeksforGeeks. The 'Output Window' on the left indicates the problem was solved successfully with 1111 test cases passed out of 1111, 1 attempt correct out of 1, 100% accuracy, 1 point scored out of 1, and a time taken of 0.28 seconds. The 'Compilation Results' tab is active. The code editor on the right shows a Java solution for finding the minimum and maximum elements in an array using a single loop. The code is as follows:

```
1 class Solution {
2     public ArrayList<Integer> getMinMax(int[] arr) {
3         // code Here
4         ArrayList<Integer> result = new ArrayList<>();
5
6         int min = arr[0];
7         int max = arr[0];
8
9         for (int i = 1; i < arr.length; i++) {
10            if (arr[i] < min) {
11                min = arr[i];
12            }
13            if (arr[i] > max) {
14                max = arr[i];
15            }
16        }
17
18        result.add(min);
19        result.add(max);
20
21        return result;
22    }
23 }
24
25 }
```

EXPERIMENT-3

The screenshot shows the GeeksforGeeks website interface for the problem "Kth Smallest Element". The user has successfully solved the problem using Java. The left sidebar displays the "Output Window" with "Compilation Results" and a "Problem Solved Successfully" message. The right pane shows the Java code for the solution.

Problem Solved Successfully ✓

Test Cases Passed: **1121 / 1121**

Attempts: Correct / Total: **1 / 1**

Accuracy: 100%

Points Scored: **4 / 4**

Time Taken: **0.74**

Your Total Score: 12 ↑

Solve Next

```
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 }
8
```

Buttons: Custom Input, Compile & Run, Submit

EXPERIMENT-4

The screenshot shows the GeeksforGeeks website interface for the problem "Union of Two Arrays". The user has successfully solved the problem using Java. The left sidebar displays the "Output Window" with "Compilation Results" and a "Problem Solved Successfully" message. The right pane shows the Java code for the solution.

Problem Solved Successfully ✓

Test Cases Passed: **1111 / 1111**

Attempts: Correct / Total: **1 / 1**

Accuracy: 100%

Points Scored: **2 / 2**

Time Taken: **1.07**

Your Total Score: 4 ↑

Solve Next

```
1 class Solution {
2     public static ArrayList<Integer> findUnion(int[] a, int[] b) {
3         // code here
4         HashSet<Integer> set = new HashSet<>();
5         for(int x:a){
6             set.add(x);
7         }
8         for(int x:b){
9             set.add(x);
10        }
11        return new ArrayList<>(set);
12    }
13 }
```

Buttons: Custom Input, Compile & Run, Submit

EXPERIMENT-5

The screenshot shows the GeeksforGeeks website interface for the problem "Largest Element in Array". The browser tabs include "JAVA EXP-1 - Google Docs", "Min and Max in Array | Practice", "Largest in Array | Practice | GeeksforGeeks", and a new tab. The URL is https://www.geeksforgeeks.org/problems/largest-element-in-array4009/0?utm_source=youtube&utm_medium=collab_striver_ytdescription&utm_campaign=striver. The page features a navigation bar with "Courses", "Tutorials", "Practice", and "Jobs" menus. The "Problem" tab is active, showing the "Output Window" with "Compilation Results". The results indicate "Problem Solved Successfully" with "Test Cases Passed: 1115 / 1115", "Attempts: Correct / Total: 1 / 1", "Accuracy: 100%", "Points Scored: 1 / 1", and "Time Taken: 0.74". The "Your Total Score" is 14. The code editor on the right shows a Java solution for finding the largest element in an array.

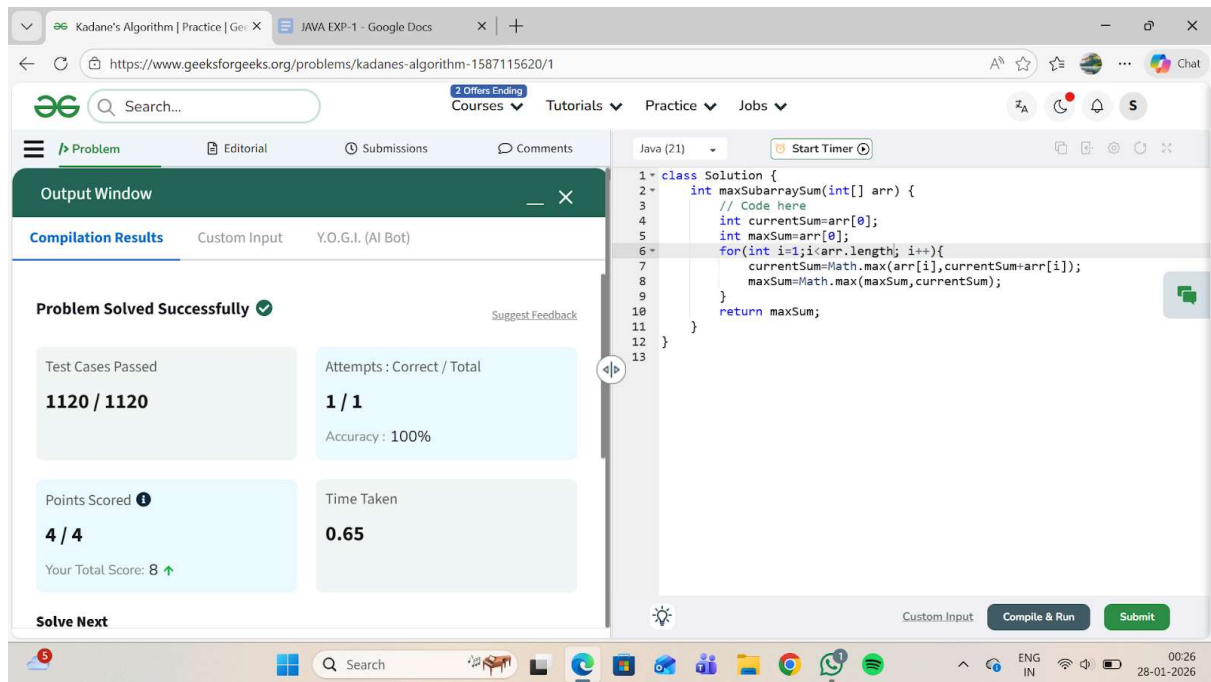
```
1 class Solution {
2     public static int largest(int[] arr) {
3         // code here
4
5         int max = arr[0];
6         for (int i = 1; i < arr.length; i++) {
7             if (arr[i] > max) {
8                 max = arr[i];
9             }
10        }
11        return max;
12    }
13 }
14
15
```

EXPERIMENT-6

The screenshot shows the GeeksforGeeks website interface for the problem "Rotate Array by One". The browser tabs include "JAVA EXP-1 - Google Docs", "Min and Max in Array | Practice", "Largest in Array | Practice | GeeksforGeeks", and "Rotate Array by One | Practice". The URL is <https://www.geeksforgeeks.org/problems/cyclically-rotate-an-array-by-one2614/1>. The page features a navigation bar with "Courses", "Tutorials", "Practice", and "Jobs" menus. The "Problem" tab is active, showing the "Output Window" with "Compilation Results". The results indicate "Problem Solved Successfully" with "Test Cases Passed: 1115 / 1115", "Attempts: Correct / Total: 1 / 1", "Accuracy: 100%", "Points Scored: 1 / 1", and "Time Taken: 1.09". The "Your Total Score" is 15. The code editor on the right shows a Java solution for rotating an array by one position.

```
1 // User function Template for Java
2
3 class Solution {
4     public void rotate(int[] arr) {
5         // code here
6         int n = arr.length;
7         int last = arr[n-1];
8         for (int i = n-1; i > 0; i--) {
9             arr[i] = arr[i-1];
10        }
11        arr[0] = last;
12    }
13 }
```

EXPERIMENT-7



The screenshot shows the GeeksforGeeks website interface for the Kadane's Algorithm problem. The left sidebar displays the 'Output Window' with 'Compilation Results' and a 'Problem Solved Successfully' message. The main content area shows the code for the 'maxSubarraySum' function in Java. The right sidebar shows the 'Test Cases Passed' (1120 / 1120), 'Attempts: Correct / Total' (1 / 1), 'Accuracy: 100%', 'Points Scored' (4 / 4), and 'Time Taken' (0.65). The bottom status bar shows the Windows taskbar with various application icons and the system clock.

Problem Solved Successfully ✓

Test Cases Passed: **1120 / 1120**

Attempts: Correct / Total: **1 / 1**

Accuracy: 100%

Points Scored: **4 / 4**

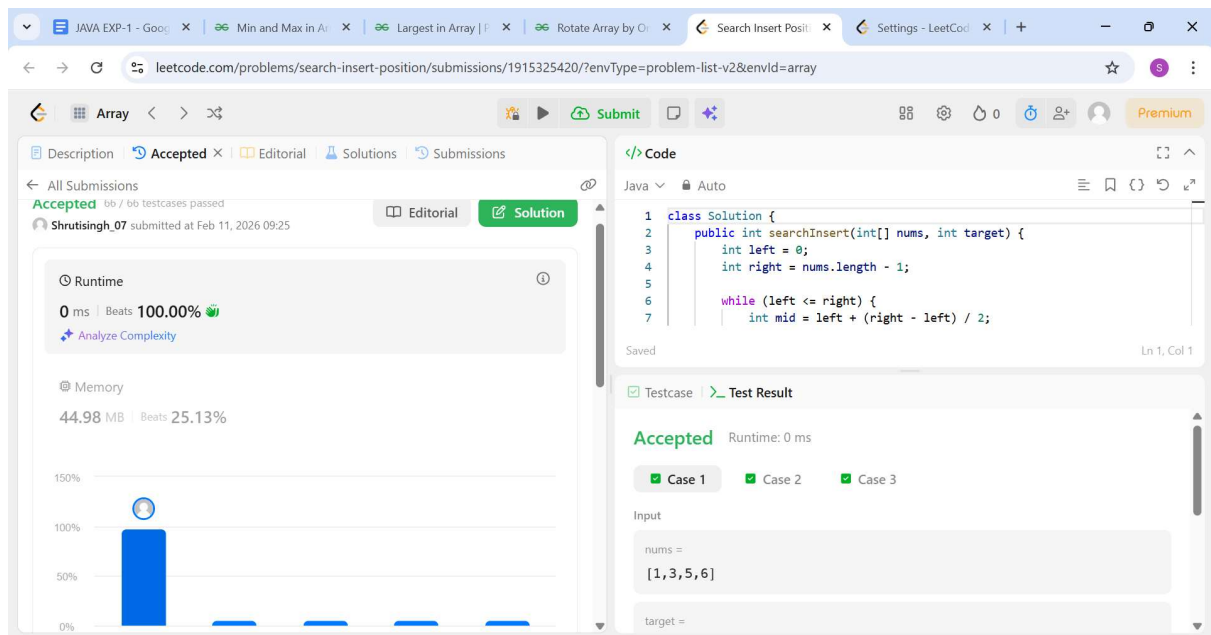
Time Taken: **0.65**

Your Total Score: 8 ↑

Solve Next

```
1 class Solution {
2     int maxSubarraySum(int[] arr) {
3         // Code here
4         int currentSum=arr[0];
5         int maxSum=arr[0];
6         for(int i=1;i<arr.length; i++){
7             currentSum=Math.max(arr[i],currentSum+arr[i]);
8             maxSum=Math.max(maxSum,currentSum);
9         }
10        return maxSum;
11    }
12 }
13
```

EXPERIMENT-8



The screenshot shows the LeetCode website interface for the Search Insert Position problem. The left sidebar displays the 'Description' and 'Accepted' status. The main content area shows the code for the 'searchInsert' function in Java. The right sidebar shows the 'Testcase' results, including 'Case 1', 'Case 2', and 'Case 3', all marked as 'Accepted'.

Accepted 06 / 06 testcases passed

Shrutisingh_07 submitted at Feb 11, 2026 09:25

Runtime

0 ms | Beats 100.00%

Memory

44.98 MB | Beats 25.13%

```
1 class Solution {
2     public int searchInsert(int[] nums, int target) {
3         int left = 0;
4         int right = nums.length - 1;
5
6         while (left <= right) {
7             int mid = left + (right - left) / 2;
```

Testcase Runtime: 0 ms

Accepted

Case 1 Case 2 Case 3

Input

nums = [1,3,5,6]

target =