

EXPERIMENT LIST FOR PROGRAMMING ABILITY AND LOGIC
BUILDING - 2

NAME : Ikshu Goel

BATCH : 2CSE5

ROLL NUMBER :

2410030774

WEEK : 19/1/26 -

25/1/26 LECTURE : 1

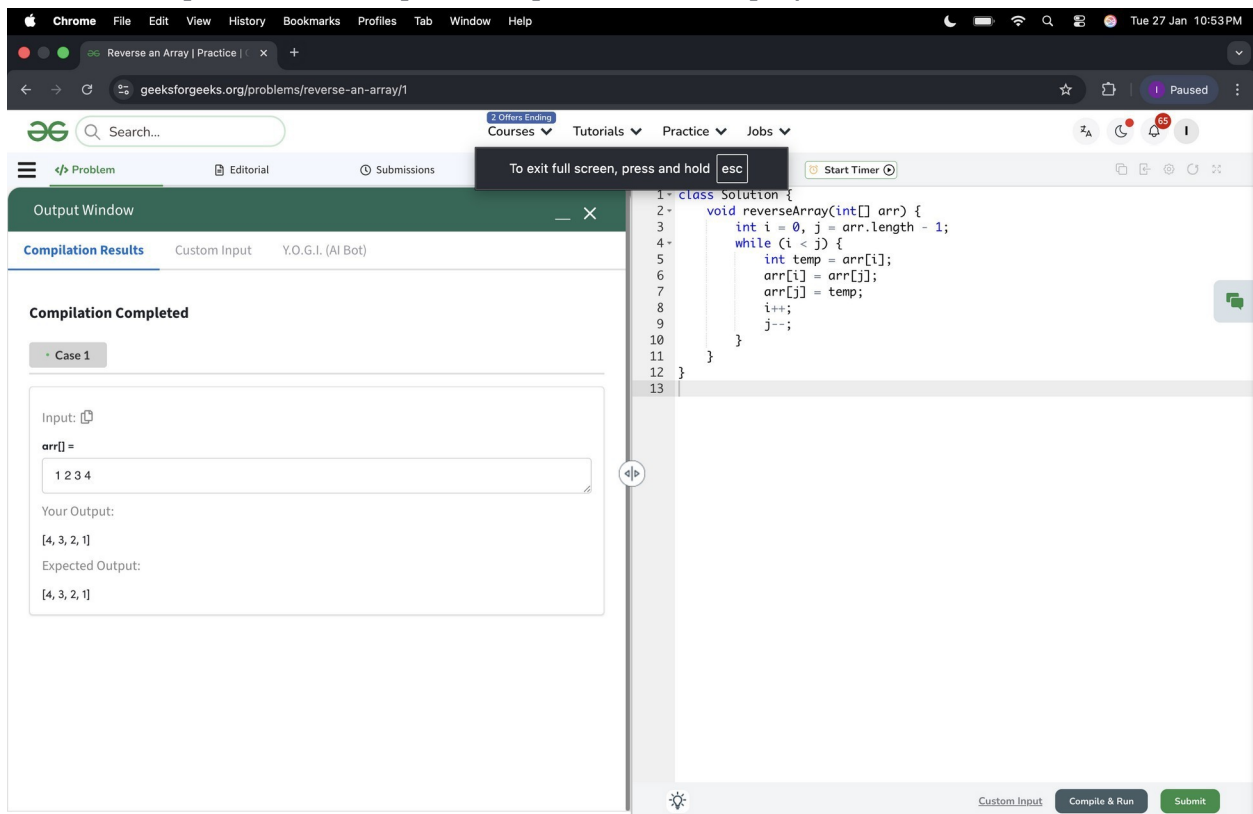
EXPERIMENT 1

You are given an array of integers `arr[]`. You have to reverse the given array. Note:

Modify the array in place.

EXAMPLE

INPUT `arr = [1, 4, 3, 2, 6, 5]` OUTPUT `[5, 6, 2, 3, 4, 1]` Explanation: The



The screenshot shows a web-based IDE interface for solving the 'Reverse an Array' problem. The browser address bar shows the URL `geeksforgeeks.org/problems/reverse-an-array/1`. The IDE has a dark theme and includes a search bar, navigation tabs (Problem, Editorial, Submissions), and a timer. The main editor displays a C++ solution for reversing an array in place using a two-pointer technique. The code is as follows:

```
1 class Solution {
2     void reverseArray(int[] arr) {
3         int i = 0, j = arr.length - 1;
4         while (i < j) {
5             int temp = arr[i];
6             arr[i] = arr[j];
7             arr[j] = temp;
8             i++;
9             j--;
10        }
11    }
12 }
13
```

On the left, the 'Output Window' shows the 'Compilation Results' for 'Case 1'. It indicates 'Compilation Completed' and displays the input and output for the test case:

Input: `arr[] =`
`1 2 3 4`
Your Output: `[4, 3, 2, 1]`
Expected Output: `[4, 3, 2, 1]`

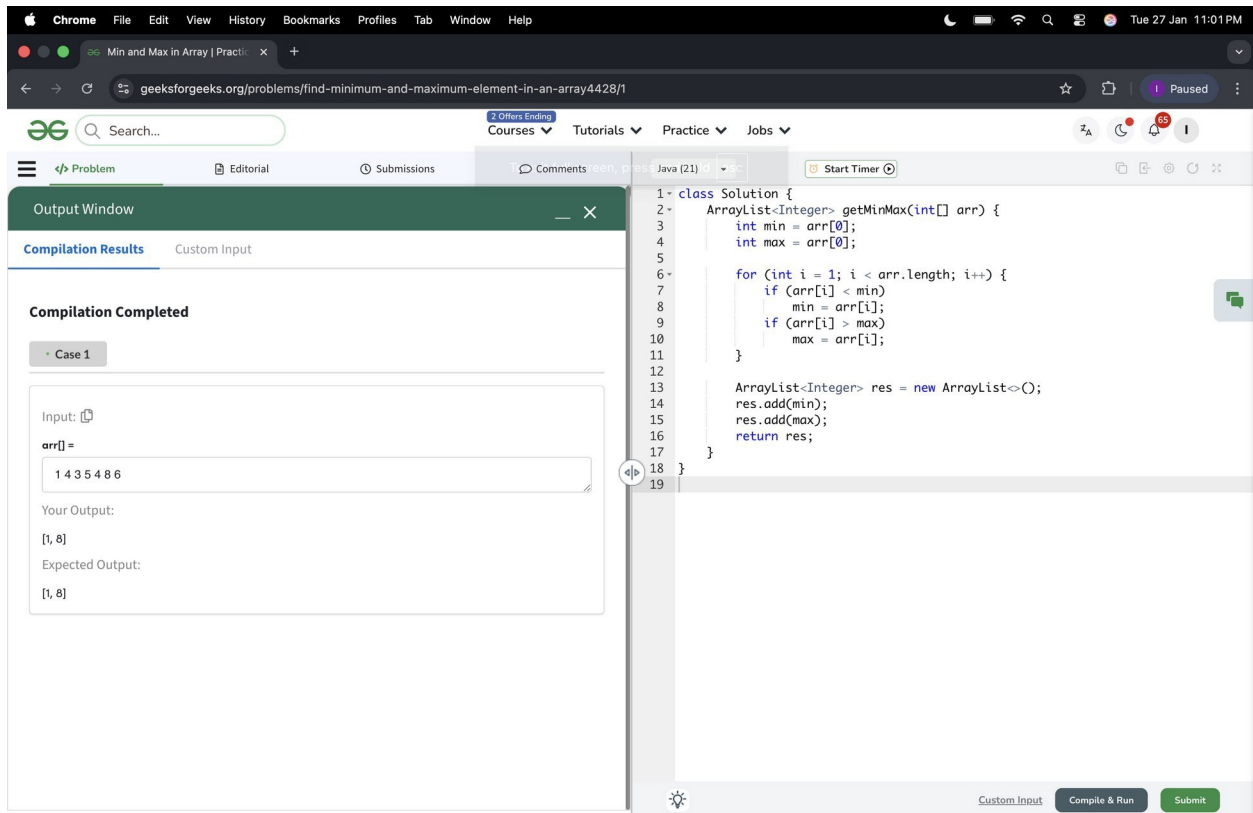
At the bottom of the IDE, there are buttons for 'Custom Input', 'Compile & Run', and 'Submit'.

elements of the array are `[1, 4, 3, 2, 6, 5]`. After reversing the array, the first element goes to the last position, the second element goes to the second last position and so on. Hence, the answer is `[5, 6, 2, 3, 4, 1]`.

EXPERIMENT 2

Given an array `arr[]`. Your task is to find the minimum and maximum elements in the array.

Examples:



The screenshot shows a web browser window with the URL `geeksforgeeks.org/problems/find-minimum-and-maximum-element-in-an-array4428/1`. The page displays a Java solution for finding the minimum and maximum elements in an array. The code is as follows:

```
1- class Solution {
2-     ArrayList<Integer> getMinMax(int[] arr) {
3-         int min = arr[0];
4-         int max = arr[0];
5-
6-         for (int i = 1; i < arr.length; i++) {
7-             if (arr[i] < min)
8-                 min = arr[i];
9-             if (arr[i] > max)
10                max = arr[i];
11         }
12
13         ArrayList<Integer> res = new ArrayList<>();
14         res.add(min);
15         res.add(max);
16         return res;
17     }
18 }
19
```

The output window on the left shows the following details:

- Compilation Results:** Custom Input
- Compilation Completed**
- Case 1:**
- Input:** `arr[] = [1 4 3 5 8 6]`
- Your Output:** `[1, 8]`
- Expected Output:** `[1, 8]`

The bottom of the page shows a `Custom Input` field, a `Compile & Run` button, and a `Submit` button.

Input: `arr[] = [1, 4, 3, 5, 8, 6]` Output: `[1, 8]` Explanation: minimum and maximum elements of array are 1 and 8.

EXPERIMENT 3

Given an integer array `arr[]` and an integer `k`, your task is to find and return the `k`th smallest element in the given array.

Note: The `k`th smallest element is determined based on the sorted order of the array.

Examples :

The screenshot shows the GeeksforGeeks online IDE interface for the 'Kth Smallest' problem. On the left, the 'Output Window' displays the 'Compilation Results' for 'Case 1'. The input is 'arr[] = 7 10 4 3 20 15' and 'k = 3'. The 'Your Output' is '7' and the 'Expected Output' is '7'. On the right, the code editor shows the following Java code:

```
1 import java.util.Arrays;
2
3 class Solution {
4     public int kthSmallest(int[] arr, int k) {
5         Arrays.sort(arr);
6         return arr[k - 1];
7     }
8 }
9
```

Input: `arr[] = [10, 5, 4, 3, 48, 6, 2, 33, 53, 10]`, `k = 4` Output: 5 Explanation: 4th smallest element in the given array is 5.

EXPERIMENT 4

You are given two arrays `a[]` and `b[]`, return the Union of both the arrays in any order.

The Union of two arrays is a collection of all distinct elements present in either of the arrays. If an element appears more than once in one or both arrays, it should be included only once in the result.

Note: Elements of `a[]` and `b[]` are not necessarily distinct. Note that, You can return the Union in any order but the driver code will print the result in sorted order only.

Examples:

Input: `a[] = [1, 2, 3, 2, 1]`, `b[] = [3, 2, 2, 3, 3, 2]` Output: `[1, 2, 3]` Explanation: Union set of both the arrays will be 1, 2 and 3.

The screenshot shows a web browser at [geeksforgeeks.org/problems/union-of-two-arrays3538/1](https://www.geeksforgeeks.org/problems/union-of-two-arrays3538/1). The interface includes a search bar, navigation links (Courses, Tutorials, Practice, Jobs), and a 'Problem' tab. The 'Output Window' on the left shows 'Compilation Results' for 'Case 1' with the following details:

Input:

a[] =
1 2 3 4 5

b[] =
1 2 3

Your Output:
[1, 2, 3, 4, 5]

Expected Output:
[1, 2, 3, 4, 5]

The main editor on the right shows a Java solution for finding the union of two arrays:

```
1- import java.util.*;
2
3- class Solution {
4-     public static ArrayList<Integer> findUnion(int[] a, int[] b) {
5-         HashSet<Integer> set = new HashSet<>();
6
7-         for (int x : a)
8-             set.add(x);
9
10        for (int x : b)
11            set.add(x);
12
13        ArrayList<Integer> res = new ArrayList<>();
14        for (int x : set)
15            res.add(x);
16
17        return res;
18    }
19 }
20
```

At the bottom right, there are buttons for 'Custom Input', 'Compile & Run', and 'Submit'.

EXPERIMENT 5

Given an array `arr[]`. The task is to find the largest element and return it. Examples:

Input: arr[] = [1, 8, 7, 56, 90]Output: 90Explanation: The largest element of the given array is 90.

Chrom...

Window Help6...90Tue 27 Jan 11 23 PM

ta ProfileBookmarkHisto Vie
ryw

Largest Array | Practice

geeksforgeeks.org/problems/largest-element-in-array4009/Q?utm_source=youtube&utm_medium=collab_striver_ytdescription&utm_campaign=largest-eleme

Paused

-aG, Q Search...

Courses vTutorials vPractice vJobs v

</>Problem

Editorial

(!) Submissions

DComments

Java(21)

StartTimer0

1 • class Solution {

2 •

3 public int largest(int[] arr) {

4 int max = arr[0];

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

6 if (arr[i] > max)

7 max = arr[i];

8 return max;

9

10

11

Output Window

X

r

Compilation Results

Custom Input

Compilation Completed

Case1

Input: (0

oml)=

1875690

Your Output:

90

Expected Output:

90

