**LAB # 06**

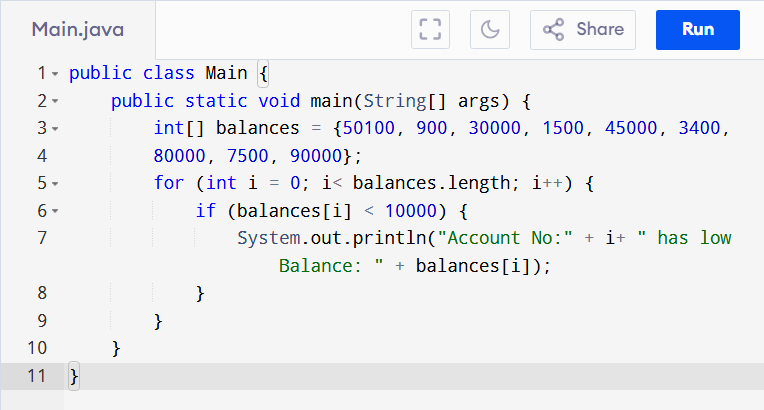
**Searching in a Linear Array**

**Lab Task**

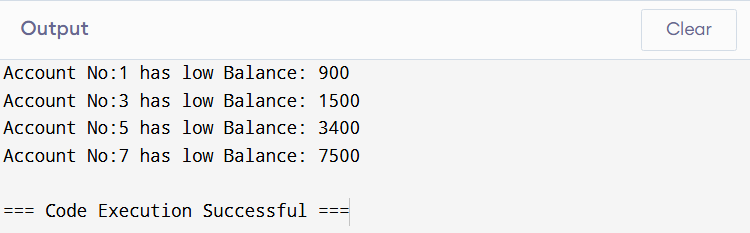
1. Declare an array of size 10 to store account balances. Initialize with values 0 to 1000000. Check all array if any value is less than 10000. Show message:

Account No. Low Balance

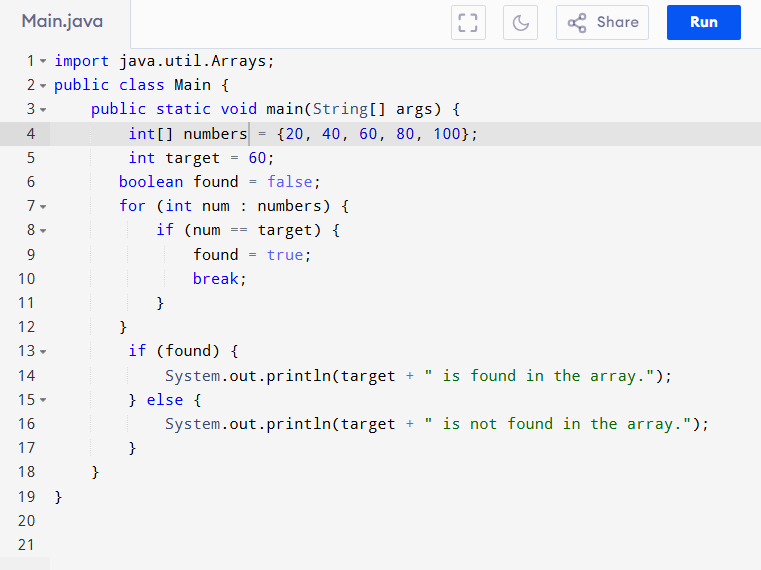
Account No. Low Balance



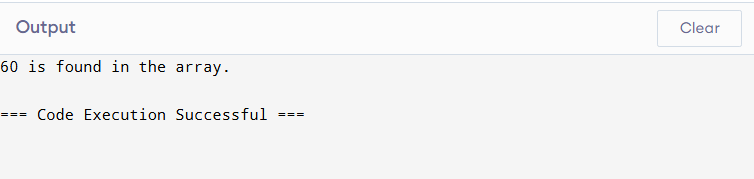
OUTPUT:



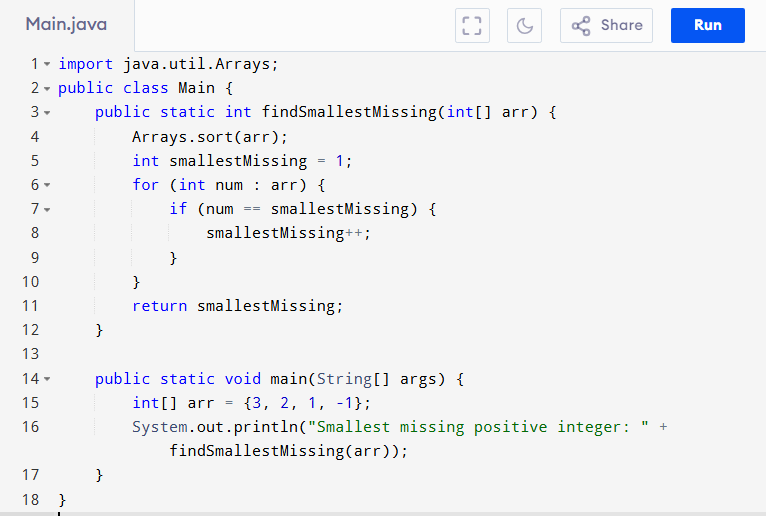
1. Write a program to search in array using Array built-in class.



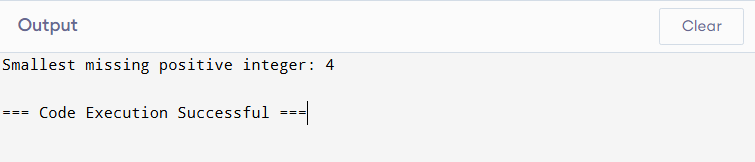
OUTPUT:

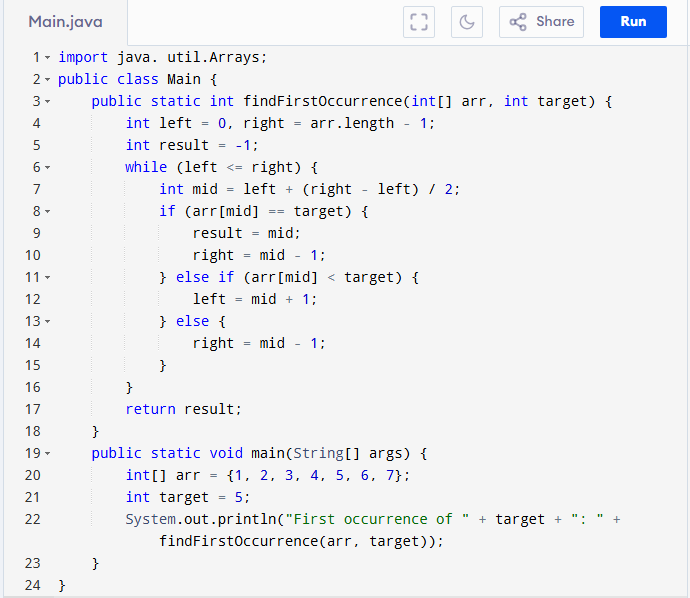


1. Given an unsorted array arr of integers, find the smallest positive integer that is **missing** from the array. You need to implement this using **binary search**. The array can contain both negative numbers and positive numbers, and you can assume that the array does not have duplicates.

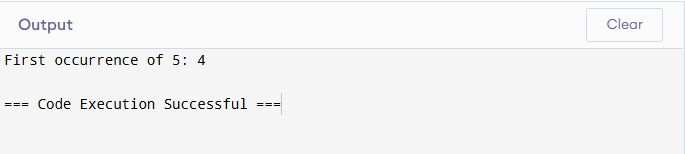


OUTPUT:



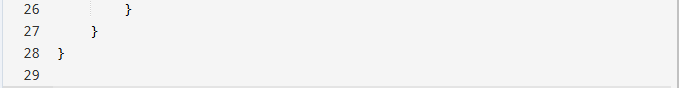
1. You are given a sorted array arr[] and a target element target. Your task is to find the **first occurrence** of the target in the array using binary search. If the target is not found, return -1. You are given a sorted array arr[] and a target element target. Your task is to find the **first occurrence** of the target in the array using binary search. If the target is not found, return -1.

OUTPUT:

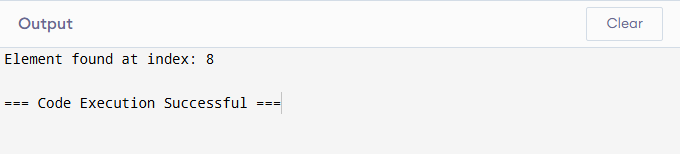


**Home Task**

1. Write a program initializing array of size 20 and search an element using binary search.



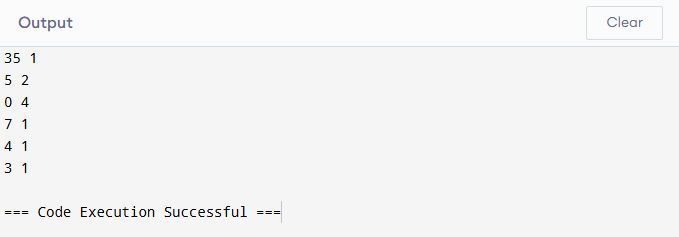
OUTPUT:



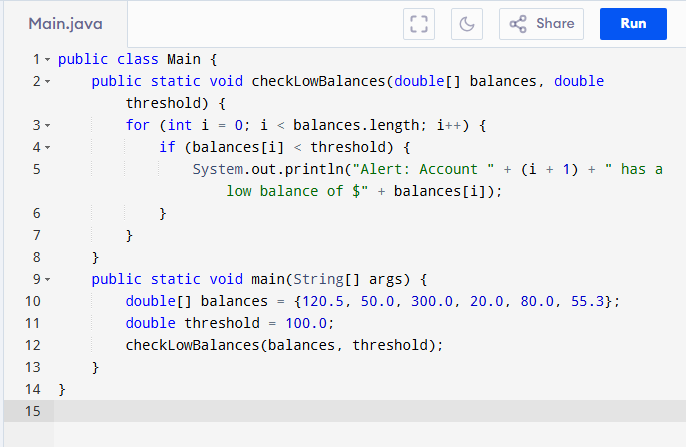
1. Write a function called occurrences that, given an array of numbers A, prints all the distinct values in A each followed by its number of occurrences.

For example, if A = (28, 1, 0, 1, 0, 3, 4, 0, 0, 3), the function should output the following five lines (here separated by a semicolon) “28 1; 1 2; 0 4; 3 2; 4 1”.



OUTPUT:

1. Assume a bank's system needs to identify accounts with critically low balances and alert the user. Test the function with various balance values to ensure it correctly identifies all accounts below the threshold.



OUTPUT:

