JAVA TASK 3

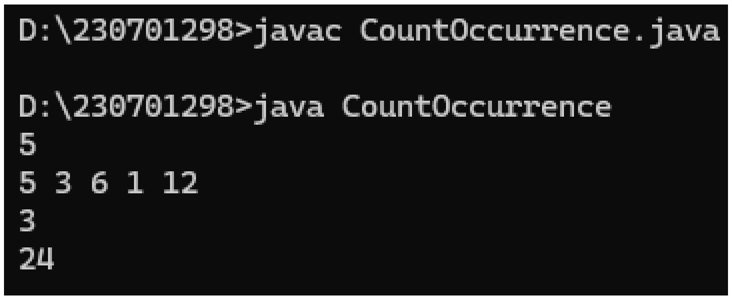
Count the occurrence

Problem Statement:

You are given an array of integers nums. You are also given an integer original which is the first number that needs to be searched for in nums.
You then do the following steps:
If original is found in nums, multiply it by two (i.e., set original = 2 * original).
Otherwise, stop the process.
Repeat this process with the new number as long as you keep finding the number.
Return the final value of original.
Input Format
First Line : The Array size n
Second Line : Array elements one in each line
Third Line : Original number
Output Format
First Line : the number
Sample Input
5
5 3 6 1 12
3
Sample Output
24
Explanation:
- 3 is found in nums. 3 is multiplied by 2 to obtain 6.
- 6 is found in nums. 6 is multiplied by 2 to obtain 12.
- 12 is found in nums. 12 is multiplied by 2 to obtain 24.
- 24 is not found in nums. Thus, 24 is returned.

Program:

```
import java.util.Scanner;

class ArrayProcessor {
    int[] nums;

    public ArrayProcessor(int[] nums) {
        this.nums = nums;
    }

    public boolean contains(int value) {
        for (int num : nums) {
            if (num == value) {
                return true;
            }
        }
        return false;
    }
}

class CountOccurrence extends ArrayProcessor {

    public CountOccurrence(int[] nums) {
        super(nums);
    }

    public int process(int original) {
        while (contains(original)) {
            original *= 2;
        }
        return original;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] nums = new int[n];
        for (int i = 0; i < n; i++) {
            nums[i] = sc.nextInt();
        }
        int original = sc.nextInt();
        CountOccurrence co = new CountOccurrence(nums);
        System.out.println(co.process(original));
    }
}
```



```
D:\230701298>javac CountOccurrence.java

D:\230701298>java CountOccurrence
5
5 3 6 1 12
3
24
```

Inventory Management

Problem Statement:

You are given an array prices where prices[i] is the price of a given stock on the ith day. You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock.
Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.
Input Format
First line : The array size : 6
Second Line : the array element s : 7 1 5 3 6 4

Output Format
First Line : 5

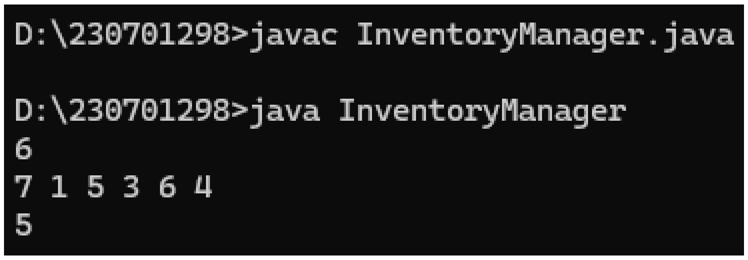
Sample Input
6
7 1 5 3 6 4

Sample Output
5

Explanation:
Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.
Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.

Program:

```
import java.util.Scanner;

class StockProcessor {
    int[] prices;

    public StockProcessor(int[] prices) {
        this.prices = prices;
    }

    public int maxProfit() {
        int minPrice = Integer.MAX_VALUE;
        int maxProfit = 0;
        for (int price : prices) {
            if (price < minPrice) {
                minPrice = price;
            } else if (price - minPrice > maxProfit) {
                maxProfit = price - minPrice;
            }
        }
        return maxProfit;
    }
}

class InventoryManager extends StockProcessor {

    public InventoryManager(int[] prices) {
        super(prices);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] prices = new int[n];
        for (int i = 0; i < n; i++) {
            prices[i] = sc.nextInt();
        }
        InventoryManager im = new InventoryManager(prices);
        System.out.println(im.maxProfit());
    }
}
```



```
D:\230701298>javac InventoryManager.java

D:\230701298>java InventoryManager
6
7 1 5 3 6 4
5
```

Sort an array of 0s, 1s and 2s

<u>Problem Statement:</u>

Given an Array of N with the elements of 0s, 1s and 2s.
Your task is to arrange the array elements in the following order.
0s followed by 1s followed 2s

Input Format
First Line : The Array size : n
Second Line: The array elements

Output Format
Print the array elements as expected.

Sample Input 1
6
{0, 1, 2, 0, 1, 2}

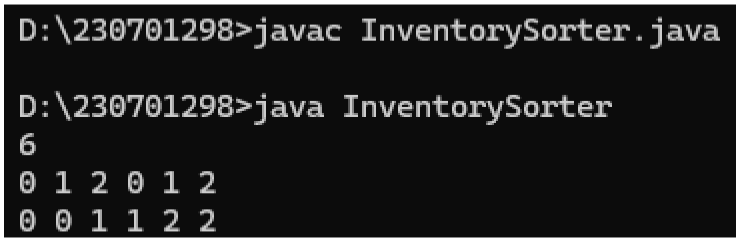
Sample Output 1
{0, 0, 1, 1, 2, 2}

Explanation: {0, 0, 1, 1, 2, 2} has all 0s first, then all 1s and all 2s in last.

Input: {0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1}
Output: {0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2}
Explanation: {0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2} has all 0s first, then all 1s and all 2s in last.

Program:

```
import java.util.Scanner;

class ArraySorter {
    int[] arr;

    public ArraySorter(int[] arr) {
        this.arr = arr;
    }

    public void sort() {
        int low = 0, mid = 0, high = arr.length - 1;
        while (mid <= high) {
            if (arr[mid] == 0) {
                swap(low++, mid++);
            } else if (arr[mid] == 1) {
                mid++;
            } else {
                swap(mid, high--);
            }
        }
    }

    private void swap(int i, int j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
}

class InventorySorter extends ArraySorter {

    public InventorySorter(int[] arr) {
        super(arr);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        InventorySorter is = new InventorySorter(arr);
        is.sort();
        for (int num : arr) {
            System.out.print(num + " ");
        }
    }
}
```



```
D:\230701298>javac InventorySorter.java

D:\230701298>java InventorySorter
6
0 1 2 0 1 2
0 0 1 1 2 2
```