53. Intersection of Three Sorted Arrays

Given three integer arrays arr1, arr2 and arr3 sorted in strictly increasing order, return a sorted array of only the integers that appeared in all three arrays.

Example 1: Input: arr1 = [1,2,3,4,5], arr2 = [1,2,5,7,9], arr3 = [1,3,4,5,8] Output: [1,5] Explanation: Only 1 and 5 appeared in the three arrays.

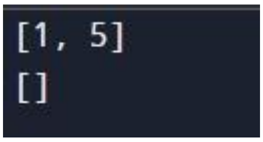
Example 2: Input: arr1 = [197,418,523,876,1356], arr2 = [501,880,1593,1710,1870], arr3 = [521,682,1337,1395,1764] Output: []

AIM: Intersection of Three Sorted Arrays

PROGRAM:

```
def arraysIntersection(arr1, arr2, arr3):
    result = []
    i = j = k = 0
    while i < len(arr1) and j < len(arr2) and k < len(arr3):
        if arr1[i] == arr2[j] == arr3[k]:
            result.append(arr1[i])
            i += 1
            j += 1
            k += 1
        elif arr1[i] < arr2[j]:
            i += 1
        elif arr2[j] < arr3[k]:
            j += 1
        else:
            k += 1
    return result
arr1 = [1, 2, 3, 4, 5]
arr2 = [1, 2, 5, 7, 9]
arr3 = [1, 3, 4, 5, 8]
print(arraysIntersection(arr1, arr2, arr3))
arr1 = [197, 418, 523, 876, 1356]
arr2 = [501, 880, 1593, 1710, 1870]
arr3 = [521, 682, 1337, 1395, 1764]
```

```
print(arraysIntersection(arr1, arr2, arr3))
```

OUTPUT: 

```
[1, 5]
[]
```

TIME COMPLEXITY: O(n)