

Gita Leoni Dwi Fripta

Kelas A

Iterable dan map data struktur

1. Buatlah sebuah program menggabungkan 2 array yang diberikan, dan jangan sampai terdapat nama yang sama di data yang sudah tergabung tadi.

Sample Test Cases

Input: ['kazuya', 'jin', 'lee'], ['kazuya', 'feng']

Output: ['kazuya', 'jin', 'lee', 'feng']

Input: ['lee', 'jin'], ['kazuya', 'panda']

Output: ['lee', 'jin', 'kazuya', 'panda']

Program tersebut yaitu menggabungkan 2 array menjadi satu, dengan hasil output menggabungkan array 1 dan array 2.

```
1 import java.util.*;
2
3 public class soal1 {
4
5     public static void main(String[] args) {
6         String[] arr1 = {"kazuya", "jin", "lee"};
7         String[] arr2 = {"kazuya", "feng"};
8
9         List<String> mergedList = new ArrayList<>(Arrays.asList(arr1));
10        for (String s : arr2) {
11            if (!mergedList.contains(s)) {
12                mergedList.add(s);
13            }
14        }
15
16        String[] mergedArr = mergedList.toArray(new String[0]);
17
18        System.out.println(Arrays.toString(mergedArr));
19    }
20 }
```

```
soal1 x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Prog
[kazuya, jin, lee, feng]

Process finished with exit code 0
```

2. Buat program sesuai dengan deskripsi di bawah. Input merupakan variable string berisi kumpulan angka. Output merupakan list / array berisi angka yang hanya muncul 1 kali pada input.

Sample Test Case

Input: "76523752"

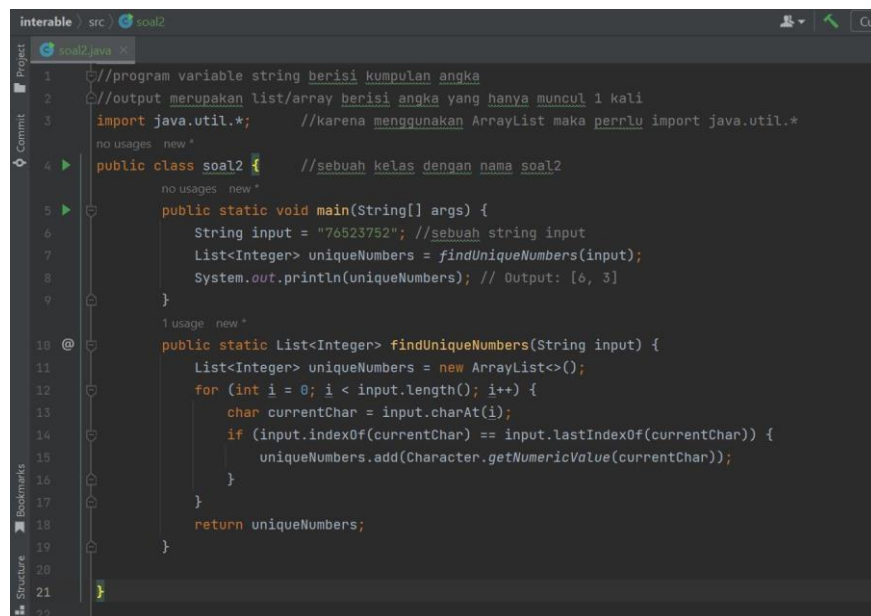
Output: [6, 3]

Input: "1122"

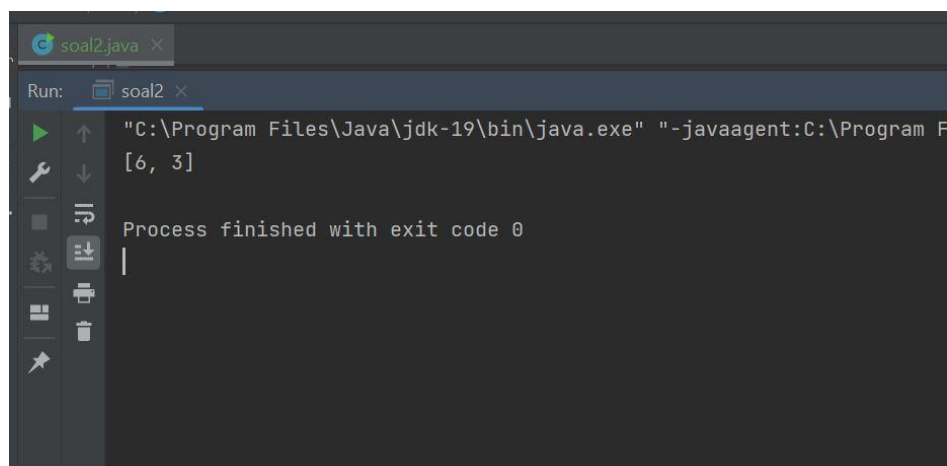
Output: []

code sample test case 1

variable string berisi kumpulan angka, merupakan array list



```
1 //program variable string berisi kumpulan angka
2 //output merupakan list/array berisi angka yang hanya muncul 1 kali
3 import java.util.*; //karena menggunakan ArrayList maka perlu import java.util.*
4 public class soal2 { //sebuah kelas dengan nama soal2
5     public static void main(String[] args) {
6         String input = "76523752"; //sebuah string input
7         List<Integer> uniqueNumbers = findUniqueNumbers(input);
8         System.out.println(uniqueNumbers); // Output: [6, 3]
9     }
10    public static List<Integer> findUniqueNumbers(String input) {
11        List<Integer> uniqueNumbers = new ArrayList<>();
12        for (int i = 0; i < input.length(); i++) {
13            char currentChar = input.charAt(i);
14            if (input.indexOf(currentChar) == input.lastIndexOf(currentChar)) {
15                uniqueNumbers.add(Character.getNumericValue(currentChar));
16            }
17        }
18        return uniqueNumbers;
19    }
20 }
21
22
```



```
Run: soal2
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program F
[6, 3]
Process finished with exit code 0
```

Sample test case2

```
soal1.java x soal2_sample2.java x soal2.java x
4 public class soal2_sample2 {
    no usages new *
5 public static void main(String[] args) {
6     String input = "1122"; //sebuah string input
7     List<Integer> uniqueNumbers = findUniqueNumbers(input);
8     System.out.println(uniqueNumbers); // Output: [6, 3]
9 }
1 usage new *
10 @ public static List<Integer> findUniqueNumbers(String input) {
11     List<Integer> uniqueNumbers = new ArrayList<>();
12     for (int i = 0; i < input.length(); i++) {
13         char currentChar = input.charAt(i);
14         if (input.indexOf(currentChar) == input.lastIndexOf(currentChar)) {
15             uniqueNumbers.add(Character.getNumericValue(currentChar));
16         }
17     }
18     return uniqueNumbers;
19 }
20 }
21 }
```

```
soal2_sample2 x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-ja
[]

Process finished with exit code 0
```

3. Given an array of sorted numbers and a target sum, find a pair in the array whose sum is equal to the given target. Write a function to return the indices of the two numbers (i.e. the pair) such that they add up to the given target.

Challenges:

Solve with linear complexity $O(n)$, **not** $O(n^2)$ if you can!

Sample Test Cases

Input: [1, 2, 3, 4, 6], target=6

Output: [1, 3]

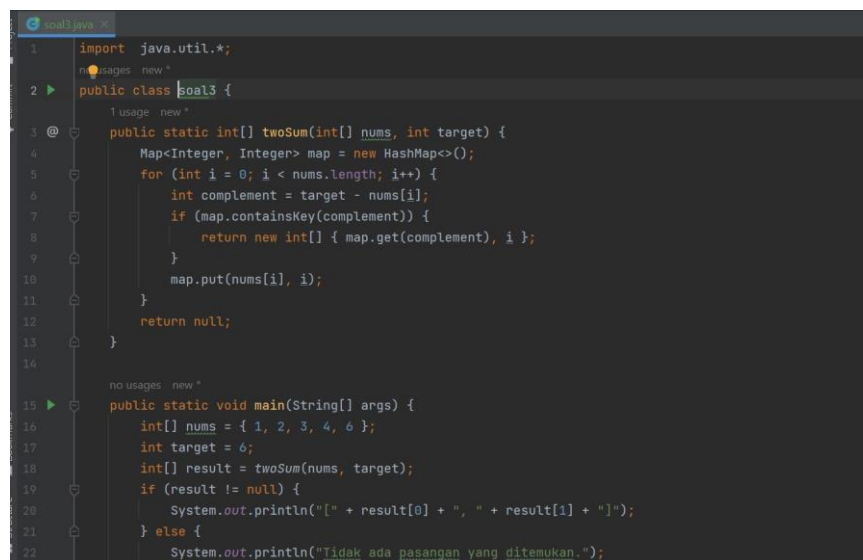
Explanation: The numbers at index 1 and 3 add up to 6: $2+4=6$

Input: [2, 5, 9, 11], target=11

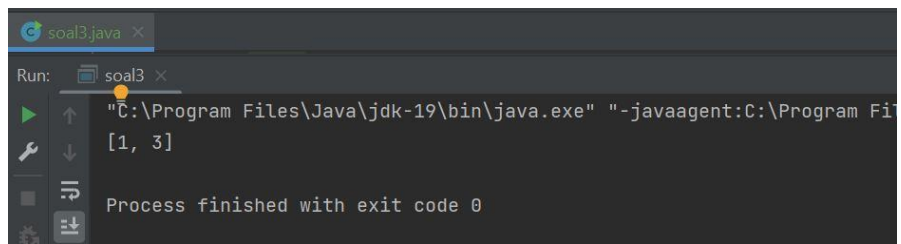
Output: [0, 2]

Explanation: The numbers at index 0 and 2 add up to 11: $2+9=11$

Sample test case 1

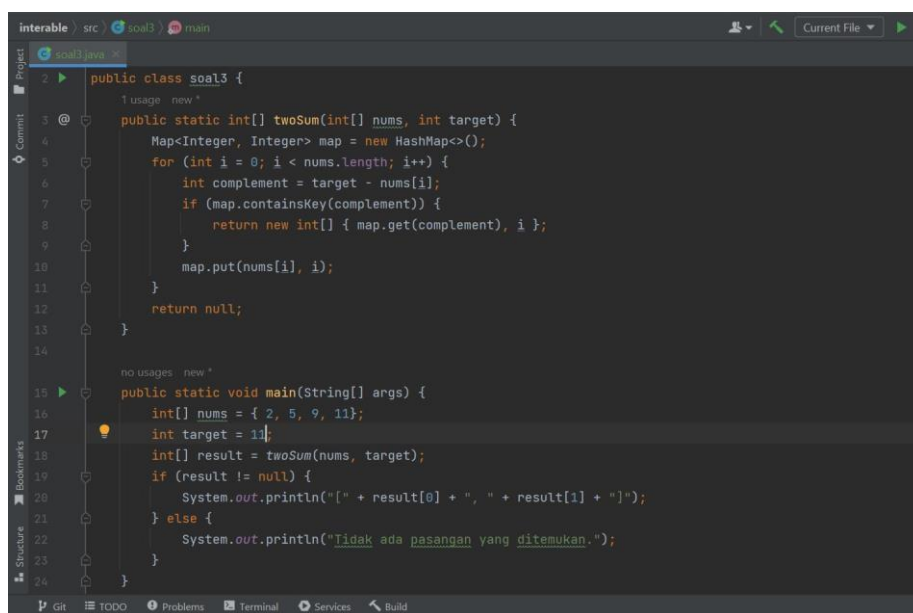


```
1 import java.util.*;
2 public class soal3 {
3     public static int[] twoSum(int[] nums, int target) {
4         Map<Integer, Integer> map = new HashMap<>();
5         for (int i = 0; i < nums.length; i++) {
6             int complement = target - nums[i];
7             if (map.containsKey(complement)) {
8                 return new int[] { map.get(complement), i };
9             }
10            map.put(nums[i], i);
11        }
12        return null;
13    }
14
15    public static void main(String[] args) {
16        int[] nums = { 1, 2, 3, 4, 6 };
17        int target = 6;
18        int[] result = twoSum(nums, target);
19        if (result != null) {
20            System.out.println "[" + result[0] + ", " + result[1] + "]";
21        } else {
22            System.out.println "Tidak ada pasangan yang ditemukan.";
23        }
24    }
25 }
```

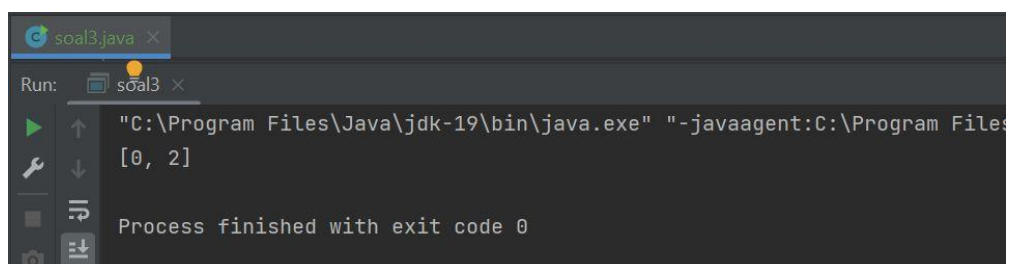


```
soal3.java x
Run: soal3 x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-19\bin\java.exe" [1, 3]
Process finished with exit code 0
```

Sample test case 2



```
interable > src > soal3 > main
soal3.java x
public class soal3 {
    1 usage: new "
    2 public static int[] twoSum(int[] nums, int target) {
    3     Map<Integer, Integer> map = new HashMap<>();
    4     for (int i = 0; i < nums.length; i++) {
    5         int complement = target - nums[i];
    6         if (map.containsKey(complement)) {
    7             return new int[] { map.get(complement), i };
    8         }
    9         map.put(nums[i], i);
    10     }
    11     return null;
    12 }
    13
    14 no usages: new "
    15 public static void main(String[] args) {
    16     int[] nums = { 2, 5, 9, 11};
    17     int target = 11;
    18     int[] result = twoSum(nums, target);
    19     if (result != null) {
    20         System.out.println "[" + result[0] + ", " + result[1] + "]";
    21     } else {
    22         System.out.println "Tidak ada pasangan yang ditemukan.";
    23     }
    24 }
```



```
soal3.java x
Run: soal3 x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-19\bin\java.exe" [0, 2]
Process finished with exit code 0
```

4. Buatlah sebuah program **ArrayUnique** yang menerima 2 parameter berupa array angka. Output adalah program adalah satu array berupa kumpulan angka di array pertama tetapi tidak memiliki duplikasi di di array kedua.

Sample Test Case

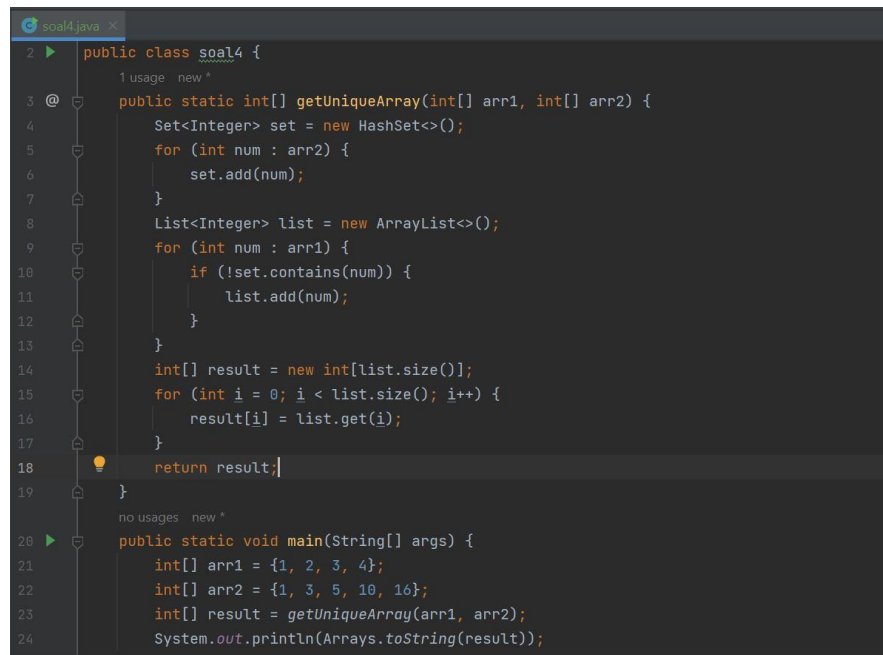
input: [1, 2, 3, 4] dan [1, 3, 5, 10, 16]

Output: [2, 4]

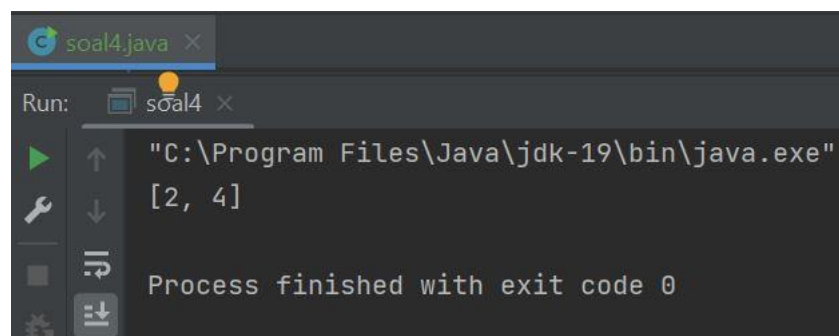
input: [3, 8] dan [2, 8]

Output: [3]

Sample test cae 1



```
2 public class soal4 {  
3     1 usage new *  
4     public static int[] getUniqueArray(int[] arr1, int[] arr2) {  
5         Set<Integer> set = new HashSet<>();  
6         for (int num : arr2) {  
7             set.add(num);  
8         }  
9         List<Integer> list = new ArrayList<>();  
10        for (int num : arr1) {  
11            if (!set.contains(num)) {  
12                list.add(num);  
13            }  
14        }  
15        int[] result = new int[list.size()];  
16        for (int i = 0; i < list.size(); i++) {  
17            result[i] = list.get(i);  
18        }  
19        return result;  
20    }  
21    no usages new *  
22    public static void main(String[] args) {  
23        int[] arr1 = {1, 2, 3, 4};  
24        int[] arr2 = {1, 3, 5, 10, 16};  
25        int[] result = getUniqueArray(arr1, arr2);  
26        System.out.println(Arrays.toString(result));  
27    }  
28 }
```



```
Run: soal4 x  
"C:\Program Files\Java\jdk-19\bin\java.exe"  
[2, 4]  
Process finished with exit code 0
```

Sample test case 2

```
soal4.java x
2 public class soal4 {
  1 usage new *
3 @ public static int[] getUniqueArray(int[] arr1, int[] arr2) {
4     Set<Integer> set = new HashSet<>();
5     for (int num : arr2) {
6         set.add(num);
7     }
8     List<Integer> list = new ArrayList<>();
9     for (int num : arr1) {
10        if (!set.contains(num)) {
11            list.add(num);
12        }
13    }
14    int[] result = new int[list.size()];
15    for (int i = 0; i < list.size(); i++) {
16        result[i] = list.get(i);
17    }
18    return result;
19 }
  no usages new *
20 public static void main(String[] args) {
21     int[] arr1 = {3,8};
22     int[] arr2 = {2,8};
23     int[] result = getUniqueArray(arr1, arr2);
24     System.out.println(Arrays.toString(result));
}
```

```
soal4.java x
Run: soal4 x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\
[3]
Process finished with exit code 0
```

5. Given an array of sorted numbers, remove all duplicates from it. You should not use any extra space; after removing the duplicates in-place return the length of the subarray that has no duplicate in it.

Sample Test Case

Input: [2, 3, 3, 3, 6, 9, 9]

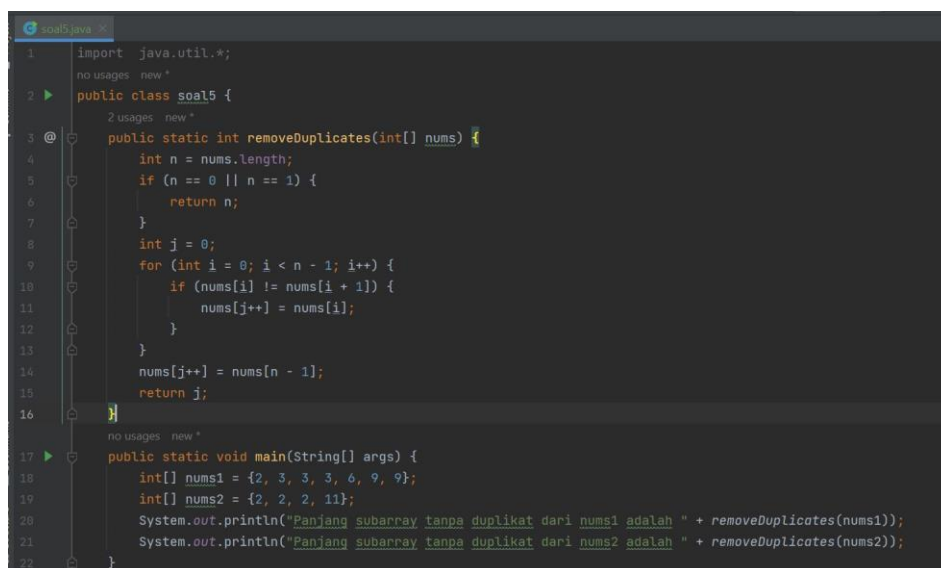
Output: 4

Explanation: The first four elements after removing the duplicates will be [2, 3, 6, 9].

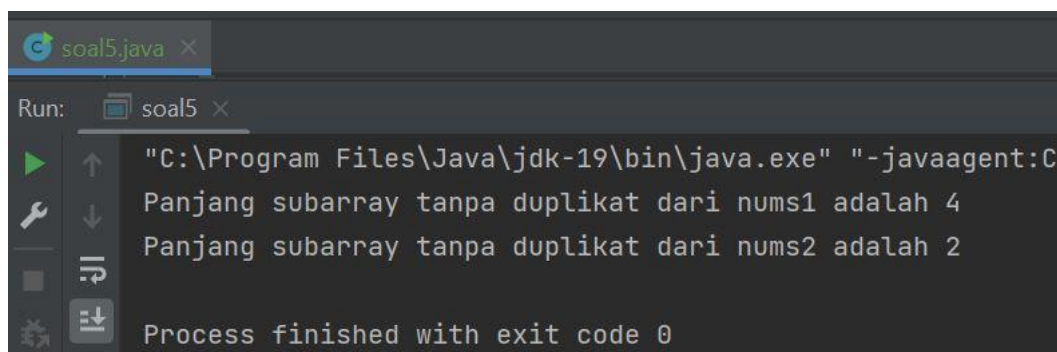
Sample Test Case

Input: [2, 2, 2, 11]

Output: 2



```
1 import java.util.*;
2 public class soal5 {
3     public static int removeDuplicates(int[] nums) {
4         int n = nums.length;
5         if (n == 0 || n == 1) {
6             return n;
7         }
8         int j = 0;
9         for (int i = 0; i < n - 1; i++) {
10             if (nums[i] != nums[i + 1]) {
11                 nums[j++] = nums[i];
12             }
13         }
14         nums[j++] = nums[n - 1];
15         return j;
16     }
17     public static void main(String[] args) {
18         int[] nums1 = {2, 3, 3, 3, 6, 9, 9};
19         int[] nums2 = {2, 2, 2, 11};
20         System.out.println("Panjang subarray tanpa duplikat dari nums1 adalah " + removeDuplicates(nums1));
21         System.out.println("Panjang subarray tanpa duplikat dari nums2 adalah " + removeDuplicates(nums2));
22     }
}
```



```
Run: soal5
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-19\bin\javaagent.jar"
Panjang subarray tanpa duplikat dari nums1 adalah 4
Panjang subarray tanpa duplikat dari nums2 adalah 2
Process finished with exit code 0
```


6. [Optional / Nilai Tambah] Given an array of positive numbers and a positive number 'k', find the maximum sum of any contiguous subarray of size 'k'.

Sample Test Case

Input: [2, 1, 5, 1, 3, 2], k=3

Output: 9

Explanation: Subarray with maximum sum is [5, 1, 3].

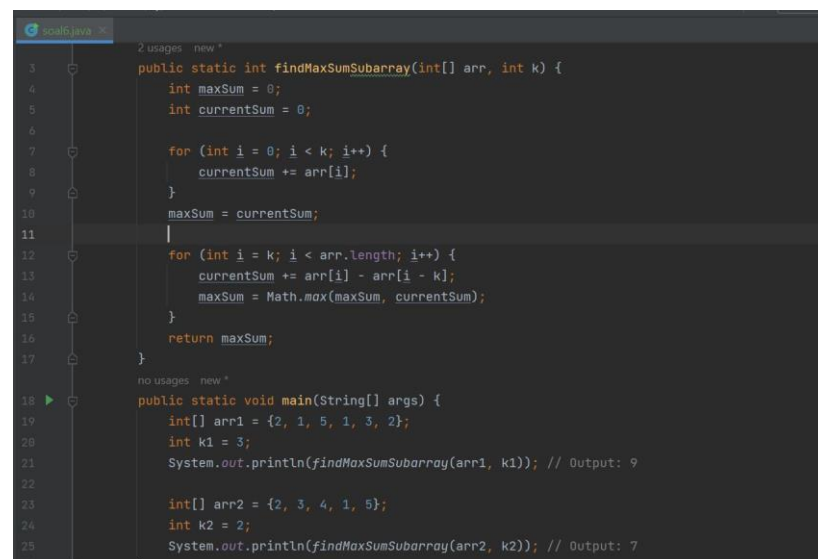
Sample Test Case

Input: [2, 3, 4, 1, 5], k=2

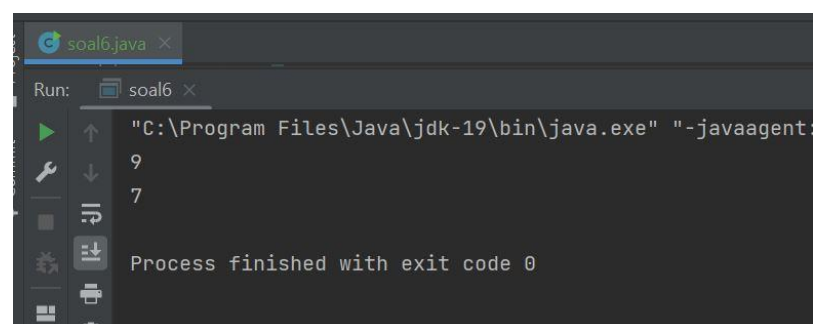
Output: 7

Explanation: Subarray with maximum sum is [3, 4].

2, 1, 5, 1, 3, 2



```
soal6.java x 2 usages new *
3 public static int findMaxSumSubarray(int[] arr, int k) {
4     int maxSum = 0;
5     int currentSum = 0;
6
7     for (int i = 0; i < k; i++) {
8         currentSum += arr[i];
9     }
10    maxSum = currentSum;
11
12    for (int i = k; i < arr.length; i++) {
13        currentSum += arr[i] - arr[i - k];
14        maxSum = Math.max(maxSum, currentSum);
15    }
16    return maxSum;
17 }
18 no usages new *
19 public static void main(String[] args) {
20     int[] arr1 = {2, 1, 5, 1, 3, 2};
21     int k1 = 3;
22     System.out.println(findMaxSumSubarray(arr1, k1)); // Output: 9
23
24     int[] arr2 = {2, 3, 4, 1, 5};
25     int k2 = 2;
26     System.out.println(findMaxSumSubarray(arr2, k2)); // Output: 7
27 }
```



```
Run: soal6 x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:
9
7
Process finished with exit code 0
```