

题目描述：

给定两个整数数组 `array1`、`array2`，数组元素按升序排列。假设从 `array1`、`array2` 中分别取出一个元素可构成一对元素，现在需要取出 `k` 对元素，并对取出的所有元素求和，计算和的最小值

注意：两对元素如果对应于 `array1`、`array2` 中的两个下标均相同，则视为同一对元素。

输入描述：

输入两行数组 `array1`、`array2`，每行首个数字为数组大小 `size` ($0 < size \leq 100$);

$0 < array1[i] \leq 1000$

$0 < array2[i] \leq 1000$

接下来一行为正整数 `k`

$0 < k \leq array1.size() * array2.size()$

输出描述：

满足要求的最小和

示例 1

输入：

```
3 1 1 2
3 1 2 3
2
```

输出：

```
4
```

说明：

用例中，需要取 2 对元素

取第一个数组第 0 个元素与第二个数组第 0 个元素组成 1 对元素 `[1,1]`;

取第一个数组第 1 个元素与第二个数组第 0 个元素组成 1 对元素 `[1,1]`;

求和为 $1+1+1+1=4$ ，为满足要求的最小和

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
```

```

import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.util.Arrays;
import java.util.PriorityQueue;
import java.util.Queue;
import java.util.StringJoiner;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        PrintWriter pw = new PrintWriter(new OutputStreamWriter(System.out));

        int[] a = Arrays.stream(br.readLine().split(" ")).skip(1).mapToInt(
            Integer::parseInt).toArray();
        int[] b = Arrays.stream(br.readLine().split(" ")).skip(1).mapToInt(
            Integer::parseInt).toArray();

        int k = Integer.parseInt(br.readLine());

        Queue<Integer> minHeap = new PriorityQueue<>();
        for (int i = 0; i < a.length; i++) {
            for (int j = 0; j < b.length; j++) {
                int sum = a[i] + b[j];
                minHeap.offer(sum);
            }
        }

        int sum = 0;
        while (k-- > 0) {
            sum += minHeap.poll();
        }

        pw.println(sum);
        pw.flush();
    }
}

```