Day 49 coding Statement : Given 2 integer arrays X and Y of same size. Consider both arrays as vectors and print the minimum scalar product (Dot product) of 2 vectors.

Sample input 1 :

4

1 2 3 4

5 6 7 8

Sample output 1 :

60

Explanation :

(4*5 + 3*6 + 2*7 + 1*8) = 60

Sample input 2 :

4

-1 -2 -3 -4

5 6 -7 -8

Sample output 2 :

-17

Explanation :

(-1*-8 + -2*-7 + -3*6 + -4*5) = -17

```java
import java.util.*;
import java.lang.*;
import java.io.*;

public class Program {
	static void swap(int arr[], int start, int end) {
		int temp = arr[start];
		arr[start] = arr[end];
		arr[end] = temp;
	}

// SpecialSort function sorts negetive numbers in array1 in ascending
```

```java
	// and positive numbers and zero in descending order
	static void SpecialSort(int vec1[], int n) {
		Arrays.sort(vec1);
		int idx = 0;
		while ((idx < n) && (vec1[idx] < 0)) {
			idx++;
		}
		int start = idx, end = n - 1;
		while (start < end) {

			swap(vec1, start, end);
			;
			start++;
			end--;
		}
	}

// Find min product and move the elements to left side of both arrays
	static int MinimumScalarProduct(int vec1[], int vec2[], int n) {
		int min, sop = 0;
		int id1 = 0, id2 = 0;
		for (int i = 0; i < n; i++) {
			min = Integer.MAX_VALUE;
			for (int j = i; j < n; j++) {
				if ((vec1[i] * vec2[j]) < min) {
					min = vec1[i] * vec2[j];
					id1 = i;
					id2 = j;
				}
			}
			sop = sop + min;
			swap(vec1, i, id1);
			swap(vec2, i, id2);
		}

		return sop;
	}

	public static void main(String[] args) throws java.lang.Exception {
		Scanner sc = new Scanner(System.in);
		int n = sc.nextInt();
		int vec1[] = new int[n];
		for (int i = 0; i < n; i++) {
			vec1[i] = sc.nextInt();
		}
		int vec2[] = new int[n];
		for (int i = 0; i < n; i++) {
			vec2[i] = sc.nextInt();
		}
		SpecialSort(vec1, n);
		System.out.print(MinimumScalarProduct(vec1, vec2, n));
	}
}
```