乱序整数序列两数之和绝对值最小题目描述：

给定一个随机的整数（可能存在正整数和负整数）数组 nums ，请你在该数组中找出两个数，其和的绝对值(|nums[x]+nums[y]|)为最小值，并返回这个两个数（按从小到大返回）以及绝对值。
每种输入只会对应一个答案。但是，数组中同一个元素不能使用两遍。

输入描述：

一个通过空格分割的有序整数序列字符串，最多 1000 个整数，且整数数值范围是 [-65535, 65535]。

输出描述：

两数之和绝对值最小值

补充说明：

示例 1

输入：

```
-1 -3 7 5 11 15
```

输出：

```
-3 5 2
```

说明：

因为 |nums[0] + nums[2]| = |-3 + 5| = 2 最小，所以返回 -3 5 2

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        int[] arrNum = getArrNum();

        int[] absArr = calcMaxAbs(arrNum[0], arrNum[1]);

        int[] result = calcNums(arrNum, absArr[0], absArr[1], absArr[2]);

        System.out.print(result[0] + " " + result[1] + " " + result[2]);
    }

    private static int[] calcMaxAbs(int val1, int val2) {
        if (val1 > val2) {
```

```
                return new int[]{Math.abs(val1 + val2), val2, val1};
            }
            return new int[]{Math.abs(val1 + val2), val1, val2};
        }

        private static int[] calcNums(int[] arrNum, int absSum, int left, int right) {
            for (int i = 0; i < arrNum.length; ++i) {
                for (int j = i + 1; j < arrNum.length; ++j) {
                    int absCurr = Math.abs(arrNum[i] + arrNum[j]);
                    if (absCurr < absSum) {
                        absSum = absCurr;
                        left = Math.min(arrNum[i], arrNum[j]);
                        right = Math.max(arrNum[i], arrNum[j]);
                    }
                }
            }
            return new int[]{left, right, absSum};
        }

        private static int[] getArrNum() {
            Scanner input = new Scanner(System.in);
            String[] s = input.nextLine().split(" ");
            int[] arrNum = new int[s.length];
            for (int i = 0; i < s.length; ++i) {
                arrNum[i] = Integer.parseInt(s[i]);
            }
            return arrNum;
        }
    }
```