https://course.acciojob.com/idle?question=b60849db-15de-464d-b6b3-c0855e531cbe

- EASY
- Max Score: 30 Points

Closest Numbers

Sorting is useful as the first step in many different tasks. The most common task is to make finding things easier, but there are other uses as well.

In this case, it will make it easier to determine which pair or pairs of elements have the smallest absolute difference between them.

Given a list of unsorted integers,`arr`, your task is to find the pair of elements that have the smallest absolute difference between them. If there are multiple pairs, find them all.

The output array must consist of all pairs in increasing order.

Input Format

The first line contains a single integers `N`.

The second line contains `N` space-separated integers describing the array `arr`.

Output Format

Print all possible pairs in the format `x1 y1 x2 y2 ... xn yn` where `xi yi` is a pair of numbers having minimum difference.

Example 1

Input

```
4
5 4 3 2
```

Output

```
2 3 3 4 4 5
```

Explanation

Here, the minimum difference is 1. Valid pairs are (2, 3), (3, 4), and (4, 5).

Example 2

Input

```
6
-20 30 -520 -470 10000 20000
```

Output

```
-520 -470 -20 30
```

Explanation

(-470) - (-520) = 30 - (-20) = 50, which is the smallest difference.

Constraints

2 <= `n` <= 200000

-10^7 <= `arr[i]` <= 10^7

All `arr[i]` are unique

Topic Tags

- Sorting

My code

// n java

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

public class Main
{
	public static void main (String[] args) throws
java.lang.Exception
	{
		//your code here
	Scanner s=new Scanner(System.in);
	int n=s.nextInt();
	int arr[]=new int[n];
	for(int i=0;i<n;i++)
	  arr[i]=s.nextInt();
	Arrays.sort(arr);
	int min=arr[1]-arr[0];
	for(int i=1;i<n-1;i++)
	  { if(min>(arr[i+1]- arr[i])) min=(arr[i+1]-arr[i]);  }
	    for(int i=0;i<n-1;i++)
	  {
	    if(min==(arr[i+1]- arr[i]))
	    { System.out.print(arr[i]+" "+arr[i+1]+" ");}

	  }

	  }
}
```