Divisors of Two Integers 1

locked

Problem Submissions Leaderboard

Recently you have received two positive integer numbers **x** and **y**. You forgot them, but you remembered a shuffled list containing all divisors of **x** (including **1** and **x**) and all divisors of **y** (including **1** and **y**). If **d** is a divisor of both numbers **x** and **y** at the same time, there are two occurrences of **d** in the list.

For example, if **x**=4 and **y**=6 then the given list can be any permutation of the list **[1,2,4,1,2,3,6]**. Some of the possible lists are: **[1,1,2,4,6,3,2]**, **[4,6,1,1,2,3,2]** or **[1,6,3,2,4,1,2]**.

Your problem is to restore suitable positive integer numbers **x** and **y** that would yield the same list of divisors (possibly in different order).

It is guaranteed that the answer exists, i.e. the given list of divisors corresponds to some positive integers **x** and **y**.

Input Format

The first line contains one integer **n** ($2 \le n \le 128$) — the number of divisors of **x** and **y**. The second line of the input contains n integers $d_1, d_2, ..., d_n$ ($1 \le d_i \le 10^4$), where d_i is either divisor of **x** or divisor of **y**. If a number is divisor of both numbers **x** and **y** then there are two copies of this number in the list.

Constraints

$2 \le n \le 128$
$1 \le d_i \le 10^4$

Output Format

Print two positive integer numbers **x** and **y** — such numbers that merged list of their divisors is the permutation of the given list of integers. It is guaranteed that the answer exists

Sample Input 0

```
10
10 2 8 1 2 4 1 20 4 5
```

Sample Output 0

```
20 8
```

Submissions: 47
Max Score: 10

Rate This Challenge:

More

Current Buffer (saved locally, editable)

Python 3

```python
import math

n = int(input())
arr = [int(x) for x in input().split()]
l = len(arr)
arr.sort()
P = arr[l-1]
divOfP = []
r = int(math.sqrt(P))+1
for i in range(1,r):
    if P % i == 0:
        if i*i == P:
            divOfP.append(i)
        else:
            divOfP.append(i)
            divOfP.append(P//i)
for x in divOfP:
    arr.remove(x)
l = len(arr)
if l == 0:
    P = Q
else:
    arr.sort()
    Q = arr[l-1]
print (P,Q)
divOfP.clear()
arr.clear()
```

Line: 27 Col: 12

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
10
10 2 8 1 2 4 1 20 4 5
```

Your Output (stdout)

```
20 8
```

Expected Output

```
20 8
```