

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($|\text{nums}[x] + \text{nums}[y]|$)为最小值，并返回这两个数（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 $|\text{nums}[0] + \text{nums}[2]| = |-3 + 5| = 2$ 最小，所以返回 -3 5 2

```
#include <iostream>
```

```
#include<sstream>
```

```
#include<iostream>
```

```
#include<istream>
```

```
#include<vector>
```

```
#include<algorithm>
```

```
using namespace std;
```

```
int main() {  
  
    vector<int> nums;  
  
    int tmp;  
  
    int minnum=65535*2;  
  
    int beginnum, endnum;  
  
    int num1, num2;  
  
    while (cin >> tmp)  
  
    {  
  
        nums.push_back(tmp);  
  
    }  
  
    sort(nums.begin(), nums.end());  
  
    beginnum = 0;  
  
    endnum = nums.size() - 1;  
  
    while (beginnum!=endnum)  
  
    {  
  
        tmp = nums[beginnum] + nums[endnum];  
  
        if (abs(tmp) < minnum)  
  
        {  
  
            minnum = abs(tmp);  
  
            num1 = nums[beginnum];  
  
            num2 = nums[endnum];  

```

```
}

else break;

if (tmp > 0) endnum--;

else if (tmp < 0) beginnum++;

else continue;

}

cout << num1 << " " << num2 << " " << minnum << endl;

}

// 64 位输出请用 printf("%lld")
```