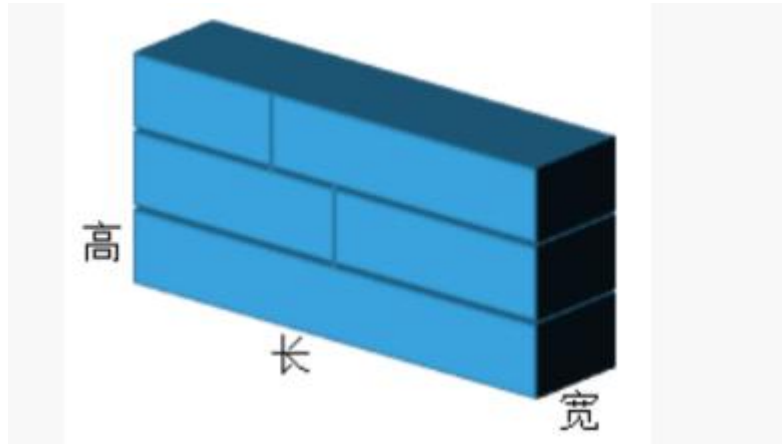


### C++-哈希表-叠积木

题目描述：有一堆长方体积木，它们的宽度和高度都相同，但长度不一。小橙想把这堆积木叠成一面墙，墙的每层可以放一个积木，也可以将两个积木拼接起来，要求每层的长度相同。若必须用完这些积木，叠成的墙最多为多少层？ 如下是叠成的一面墙的图示，积木仅按宽和高所在的面进行拼接。



输入描述：输入为一行，为各个积木的长度，数字为正整数，并由空格分隔。积木的数量和长度都不超过 5000。

输出描述：输出一个数字，为墙的最大层数，如果无法按要求叠成每层长度一致的墙，则输出-1。

示例

展开

示例1

输入：3 6 6 3

输出：3

说明：可以每层都是长度3和6的积木拼接起来，这样每层的长度为9，层数为2；也可以其中两层直接用长度6的积木，两个长度3的积木拼接为一层，这样层数为3，故输出3。

示例2

输入：1 4 2 3 6

输出：-1

说明：无法用这些积木叠成每层长度一致的墙，故输出-1。

```
1 #include <iostream>
2 #include <vector>
3 #include <cmath>
```

```
4 #include <algorithm>
5 #include <map>
6 using namespace std;
7
8 vector<int> strToVec(const string& s) {
9     string sub;
10    vector<int> v;
11    for (int i = 0; i < s.size(); i++) {
12        const char& c = s[i];
13        if (isdigit(c)) {
14            sub.append(1, c);
15        }
16        else {
17            v.push_back(stoi(sub));
18            sub.clear();
19        }
20    }
21    v.push_back(stoi(sub));
22    sub.clear();
23
24    return std::move(v);
25 }
26
27 bool check(const vector<int> v, const int& l){
28
29     if(l < v[v.size()-1]){
30         return false;
31     }
32 }
```

```

32
33     map<int,int> need;
34
35     for(int i=0;i<v.size();i++){
36         const int& t = v[i];
37         if(l==t){
38             continue;
39         }
40
41         auto p = need.find(t);
42         if(p!=need.end()&&p->second!=0){//若存在need[t]且need[t]!=0
43             need[t]--;
44         }else{
45             need[l-t]++;//
46         }
47     }
48
49     bool flag = true;
50
51     auto p = need.begin();
52     while(p!=need.end()){
53         if(p->second!=0){
54             flag = false;
55             break;
56         }
57         p++;
58     }
59

```

```

60
61     return flag;
62
63 }
64
65 int main() {
66
67     string s;
68     getline(cin,s);
69
70     vector<int> v = strToVec(s);
71     sort(v.begin(),v.end());
72
73     int sum = 0;
74     for(int i=0;i<v.size();i++){
75         sum += v[i];
76     }
77
78     int k = pow(sum,0.5);
79     int level = -1;
80     for(int i=2;i<=k;i++){//每层长i
81         if(sum%i!=0){
82             continue;
83         }else if(check(v,i)){
84             level = level < sum / i ? sum / i : level;
85         }else if(check(v,sum/i)){
86             level = level < i ? i : level;
87         }
88     }
89
90     cout<<level;
91

```

```

91

```

```

92 }
93 // 64 位输出请用 printf("%lld")

```