

找终点

题目描述：

给定一个正整数数组，设为 `nums`，最大为 `100` 个成员，求从第一个成员开始，正好走到数组最后一个成员，所使用的最少步骤数。

要求：

- 1、第一步必须从第一元素开始，且  $1 \leq \text{第一步的步长} < \text{len}/2$ ；（`len` 为数组的长度，需要自行解析）。
- 2、从第二步开始，只能以所在成员的数字走相应的步数，不能多也不能少，如果目标不可达返回 `-1`，只输出最少的步骤数量。
- 3、只能向数组的尾部走，不能往回走。

输入描述：

由正整数组成的数组，以空格分隔，数组长度小于 `100`，请自行解析数据数量。

输出描述：

正整数，表示最少的步数，如果不存在输出 `-1`

补充说明：

示例 1

输入：

7 5 9 4 2 6 8 3 5 4 3 9

输出：

2

说明：

第一步： 第一个可选步长选择 **2**，从第一个成员 **7** 开始走 **2** 步，到达 **9**；第二步： 从 **9** 开始，经过自身数字 **9** 对应的 **9** 个成员到最后。

示例 **2**

输入：

1 2 3 7 1 5 9 3 2 1

输出：

-1

说明：

```
#include <iostream>
```

```
#include <string>
```

```
#include <vector>
```

```
using namespace std;
```

```
int solve(vector<int> nums, int first) {
```

```
    int ans = 1;
```

```
    while (first < nums.size() - 1) {
```

```
        first += nums[first];
```

```
        ans++;
```

```
    }
```

```
    if (first == nums.size() - 1)
```

```
        return ans;
```

```
    return 999;
```

```
}
```

```
int main() {
```

```
    string str;
```

```
    while (getline(cin, str)) {
```

```
        vector<int> nums;
```

```
        int start = 0, pos, ans = 999;
```

```
        while (str.find(' ', start) != -1) {
```

```
            pos = str.find(' ', start);
```

```
            nums.push_back(stoi(str.substr(start, pos)));
```

```
            start = pos + 1;
```

```
        }
```

```
        pos = str.find(' ', start);
```

```
        nums.push_back(stoi(str.substr(start, pos)));
```

```
        for (int i = 1; i < nums.size() / 2; ++i) {
```

```
            ans = ans < solve(nums, i) ? ans : solve(nums, i);
```

```
        }
```

```
        if (ans == 999)
```

```
            cout << "-1" << endl;
```

```
        else
```

```
            cout << ans << endl;
```

```
    }
```

```
    return 0;
```

}