

c++-数组滑窗-有 N 个正整数组成的一个序列

题目描述：

有 N 个正整数组成的一个序列。给定整数 sum，求长度最长的连续子序列，使他们的和等于 sum，返回此子序列的长度，如果没有满足要求的序列，返回-1。

输入描述：

序列：1,2,3,4,2

sum：6

输出描述：

序列长度：3

补充说明：

输入序列仅由数字和英文逗号构成，数字之间采用英文逗号分隔；

序列长度：1 <= N <= 200；

输入序列不考虑异常情况，由题目保证输入序列满足要求。

示例 1

输入：

1,2,3,4,2

6

输出：

3

说明：

解释：1,2,3 和 4,2 两个序列均能满足要求，所以最长的连续序列为 1,2,3，因此结果为 3

示例 2

输入：

1,2,3,4,2

20

输出：

-1

说明：

解释：没有满足要求的子序列，返回-1

```
#include <iostream>
```

```
#include <cstring>
```

```
#include <vector>
```

```
using namespace std;
```

```
//输入：正数字序列、目标和
```

```
//输出：满足规则的子序列长度
```

```
//规则：
```

```
//连续+子序列+和为目标和
```

```
//求最长序列长度，没有返回-1
```

```
vector<int> v;
```

```

long long target;
int main() {
    //处理输入
    string str;
    getline(cin, str);
    str += ",";
    cin >> target;

    string temp = "";
    for (int i = 0; str[i]; i++) {
        if (str[i] == ',') {
            v.emplace_back(stoi(temp));
            temp = "";
        } else temp += str[i];
    }
    int len = v.size();

    int ans = -1;

    int l = 0, r = 0;
    int tlen = 0;
    while (l < len) {

        while (r < len && tlen < target) tlen += v[r++];
        if (tlen == target) ans = max(ans, r - l);
        tlen -= v[l++];

        //cout << ans << " " << endl;
    }
    cout << ans;

}
// 64 位输出请用 printf("%lld")

```