

最长连续子序列

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： $1 \leq N \leq 200$ ；

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

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 <vector>
```

```
using namespace std;
```

```
vector<int> getData(int &len, int &sum) {  
    vector<int> dp(1, 0);  
    do {  
        int tmp;  
        scanf("%d", &tmp);  
        dp.push_back(dp.back() + tmp);  
        len++;  
    } while (getchar() != '\n');  
    scanf("%d", &sum);  
    return dp;  
}
```

```
int calcVal(vector<int> &dp, int len, int sum) {  
    int maxLen = 0;  
    for (int i = 1; i < len; i++) {  
        for (int j = 0; j < i; j++) {  
            if (dp[i] - dp[j] == sum) {  
                maxLen = maxLen < i - j ? i - j : maxLen;  
            }  
        }  
    }  
    return maxLen;  
}
```

```
int main() {  
    int len = 1, sum;  
    vector<int> dp = getData(len, sum);  
    int maxLen = calcVal(dp, len, sum);  
    maxLen ? cout << maxLen << endl : cout << -1 << endl;
```

```
    return 0;  
}
```