

C++-题目描述:

一贫如洗的樵夫阿里巴巴在去砍柴的路上，无意中发现了强盗集团的藏宝地，藏宝地有编号从 $0 \sim N$ 的箱子，每个箱子上面贴有一个数字。

阿里巴巴念出一个咒语数字 $k(k < N)$ ，找出连续 k 个宝箱数字和的最大值，并输出该最大值。

输入描述:

第一行输入一个数字字串，数字之间使用逗号分隔，例如: `2,10,-3,-8,40,5`

字串中数字的个数 ≥ 1 ， ≤ 100000 ；每个数字 ≥ -10000 ， ≤ 10000 ；

第二行输入咒语数字，例如: `4`，咒语数字大小小于宝箱的个数

输出描述:

连续 k 个宝箱数字和的最大值，例如: `39`

补充说明:

示例 1

输入:

`2,10,-3,-8,40,5`

`4`

输出:

`39`

说明:

示例 2

输入:

`8`

`1`

输出：

8

说明：

```
#include <iostream>
#include <vector>
#include <stack>
using namespace std;
```

```
// 2,10,-3,-8,40,5
```

```
// 4
```

```
vector<int> split(string str) {
    int left = 0;
    vector<int> ret;
    for (int i = 0; i < str.size(); ++i) {
        if (str[i] == ',') {
            int num = atoi(str.substr(left, i - left + 1).c_str());
            ret.push_back(num);
            left = i + 1;
        }
    }
    int num = atoi(str.substr(left).c_str());
    ret.push_back(num);
    return ret;
}
```

```
int solution(vector<int> &arr, int n) {
    stack<int> stk;
    stk.push(arr[0]);
    for (int i = 1; i < arr.size(); ++i) {
        arr[i] += arr[i - 1];
    }
    int ret = arr[n - 1];
    for (int i = n; i < arr.size(); ++i) {
        ret = max(ret, arr[i] - arr[i - n]);
    }
    return ret;
}
```

```
int main() {
    const int sz = 1000000;
    char str[sz];
```

```
cin.getline(str,sz);
auto list = split(str);
int n;
cin >> n;
cout << solution(list, n) <<endl;
return 0;
}
```