

[问答题]

题目描述

以下函数用于找到整数矩阵 `matrix` 中，元素之和最大的 `n` 行 `m` 列的子矩阵的元素之和。请指出程序代码中错误的地方（问题不止一处，请尽量找出所有你认为错误的地方），并在不新增代码行的情况下将问题修复。

```
1 int maxSubmatrixSum(std::vector<std::vector<int>> matrix,
2     int n, int m) {
3     int base_sum;
4     for (int i = 0; i < n; i++){
5         for (int j = 0; j < m; j++){
6             base_sum += matrix[i][j];
7         }
8     }
9     int result = 0;
10    for (int i = 0; i + n < matrix.size(); i++) {
11        if(i > 0){
12            for (int y = 0; y < m; y++){
13                base_sum += matrix[i + n][y] - matrix[i - 1][y];
14            }
15        }
16        int real_sum = base_sum;
17        if (real_sum > result) {
18            result = real_sum;
19        }
20        for (int j = 0; j + m < matrix.size(); j++) {
21            for (int x = 0; x < n; x++) {
22                real_sum += matrix[x][j + m] - matrix[x][j - 1];
23            }
24            if (real_sum > result) {
25                result = real_sum;
26            }
27        }
28    }
29    return result;
30 }
```

[编程题] 编程题 1

时间限制：1 秒

空间限制：65536K

有一个推箱子的游戏，一开始的情况如下图:

```

.....
.....E.....
..###.....#####
#####.....#####
.....
.....S.....#####
.....0.....
.....

```

上图中, '.' 表示可到达的位置, '#' 表示不可到达的位置, 其中 S 表示你起始的位置, 0 表示初始箱子的位置, E 表示预期箱子的位置, 你可以走到箱子的上下左右任意一侧, 将箱子向另一侧推动。如下图将箱子向右推动一格;

```
..S0.. -> ...S0.
```

注意不能将箱子推动到 '#' 上, 也不能将箱子推出边界;

现在, 给你游戏的初始样子, 你需要输出最少几步能够完成游戏, 如果不能完成, 则输出-1。

输入描述:

第一行为 2 个数字, n, m , 表示游戏盘面大小有 n 行 m 列 ($5 < n, m < 50$);

后面为 n 行字符串, 每行字符串有 m 字符, 表示游戏盘面;

输出描述:

一个数字, 表示最少几步能完成游戏, 如果不能, 输出-1;

输入例子 1:

```
3 6
```

```
.S#...E
```

```
.#.0..
```

.....

输出例子 1:

11

[编程题] 编程题 2

时间限制：1 秒

空间限制：65536K

有 n 个房间，现在 i 号房间里的人需要被重新分配，分配的规则是这样的：先让 i 号房间里的人全都出来，接下来按照 $i+1, i+2, i+3, \dots$ 的顺序依次往这些房间里放一个人， n 号房间的的下一个房间是 1 号房间，直到所有的人都被重新分配。

现在告诉你分配完后每个房间的人数以及最后一个人被分配的房间号 x ，你需要求出分配前每个房间的人数。数据保证一定有解，若有多解输出任意一个解。

输入描述:

第一行两个整数 n, x ($2 \leq n \leq 10^5, 1 \leq x \leq n$)，代表房间数量以及最后一个人被分配的房间号；第二行 n 个整数 a_i ($0 \leq a_i \leq 10^9$)，代表每个房间分配后的人数。

输出描述:

输出 n 个整数，代表每个房间分配前的人数。

输入例子 1:

3 1

6 5 1

输出例子 1:

4 4 4

[问答题]

题目描述

【简答题】

简述 Activity、Window、WindowManager、View、ViewRootImpl 的作用和相互之间的关系。

[问答题]

题目描述

【设计题】

App 发展到一定程度时，页面越来越多，工程越来越大，合作开发的人也越来越多，这时就可能需要引入路由系统，实现模块间的解耦。

请设计一个路由系统，使得 app 内页面的跳转就像浏览器访问网页一样易于管理和解耦。