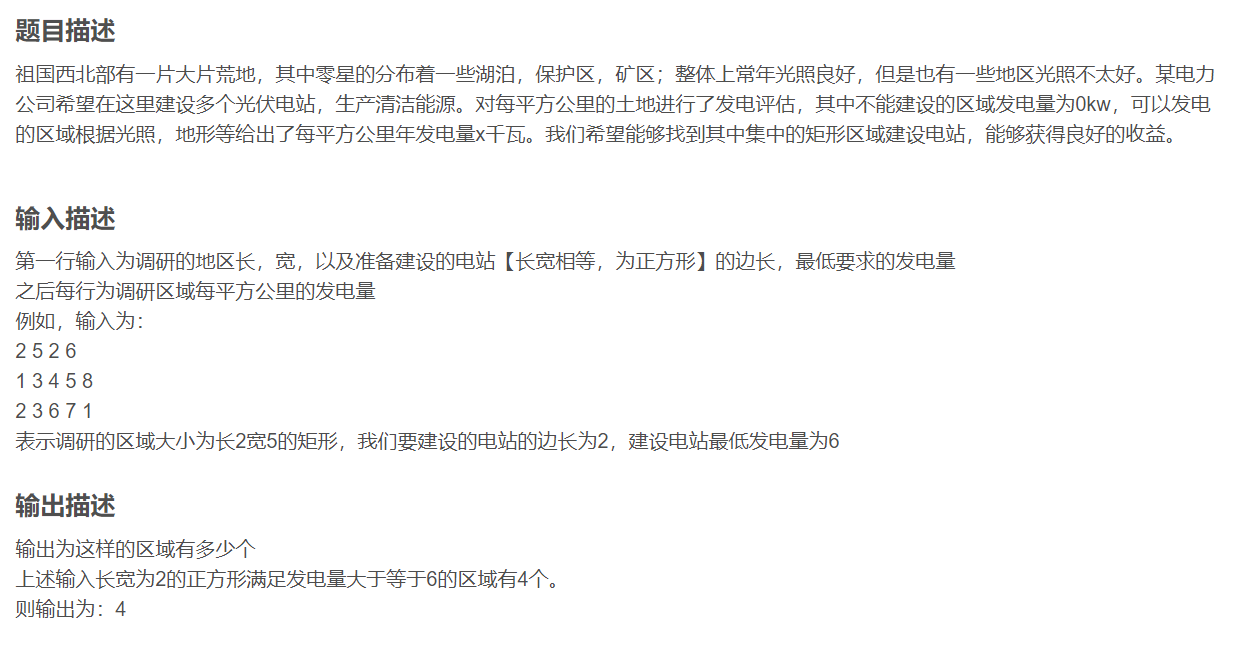
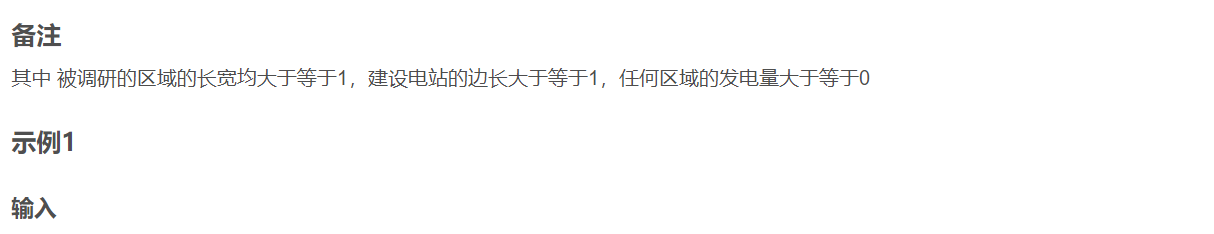
# **E卷-光伏场地建设规划[100分]（ Java | Python3 | C++ | C语言 | JsNode | Go ）**





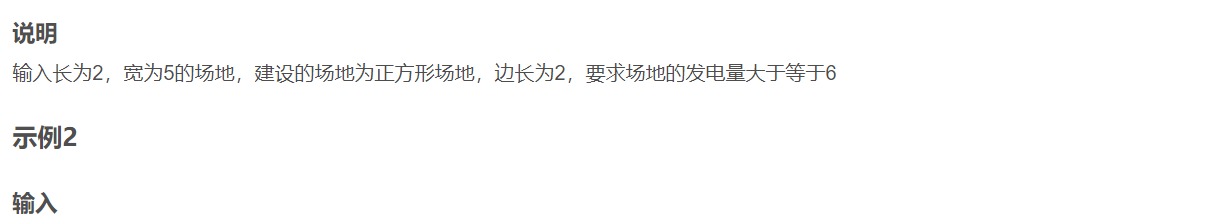
2 5 2 6

1 3 4 5 8

2 3 6 7 1



4



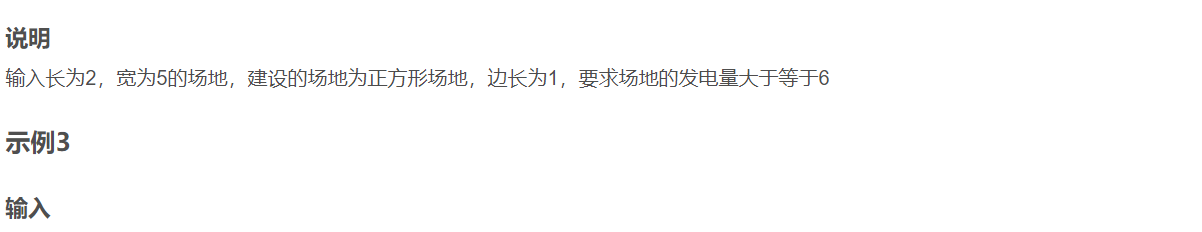
2 5 1 6

1 3 4 5 8

2 3 6 7 1



3



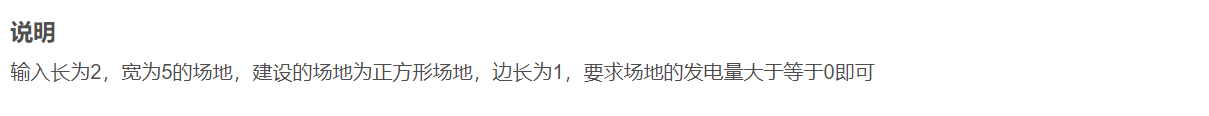
2 5 1 0

1 3 4 5 8

2 3 6 7 1



10





import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// 输入区域的长、宽，电站边长，最低发电量

int r = scanner.nextInt();

int c = scanner.nextInt();

int n = scanner.nextInt();

int power = scanner.nextInt();

// 创建并读取发电量矩阵数组

int[][] arr = new int[r][c];

for (int i = 0; i < r; ++i) {

for (int j = 0; j < c; ++j) {

arr[i][j] = scanner.nextInt();

}

}

int cnt = 0;

// 遍历可能的电站位置

for (int i = 0; i <= r - n; ++i) {

for (int j = 0; j <= c - n; ++j) {

int tot = 0;

// 计算电站区域内的总发电量

for (int k = 0; k < n; ++k) {

for (int s = 0; s < n; ++s) {

tot += arr[i + k][j + s];

}

}

// 判断是否满足最低发电量要求

if (tot >= power) {

cnt++;

}

}

}

// 输出满足条件的区域数量

System.out.println(cnt);

scanner.close();

}

}



# 导入sys以便从控制台获取输入

import sys

# 从控制台读取输入

input = sys.stdin.read

def main():

# 读取输入行并拆分为整数

data = input().strip().split()

r = int(data[0])

c = int(data[1])

n = int(data[2])

power = int(data[3])

# 创建一个2D数组来存储发电量数据

arr = []

index = 4 # 从第5个数据开始是矩阵的数据

for i in range(r):

row = list(map(int, data[index : index + c]))

arr.append(row)

index += c

cnt = 0

# 遍历所有可能的电站位置

for i in range(r - n + 1):

for j in range(c - n + 1):

tot = 0

# 计算这个n\*n区域内的发电量总和

for k in range(n):

for s in range(n):

tot += arr[i + k][j + s]

# 检查总发电量是否达到要求

if tot >= power:

cnt += 1

# 输出满足条件的电站区域数量

print(cnt)

import sys

if \_\_name\_\_ == "\_\_main\_\_":

main()



#include <iostream>

using namespace std;

int main() {

// 定义二维数组来存储发电量数据

int arr[100][100];

int r, c, n,

power; // 定义调研区域的长r，宽c，正方形边长n和最低发电量power

// 读取输入参数

cin >> r >> c >> n >> power;

// 读取发电量数据

for (int i = 0; i < r; ++i) {

for (int j = 0; j < c; ++j) {

cin >> arr[i][j];

}

}

int cnt = 0; // 计数器，用于统计满足条件的区域数

// 遍历可能的正方形区域

for (int i = 0; i < r - n + 1; ++i) {

for (int j = 0; j < c - n + 1; ++j) {

int tot = 0; // 局部变量，用于存储当前正方形区域的发电量总和

// 计算子矩阵的总发电量

for (int k = 0; k < n; ++k) {

for (int s = 0; s < n; ++s) {

tot += arr[i + k][j + s];

}

}

// 检查总发电量是否符合条件

if (tot >= power) {

cnt++;

}

}

}

// 输出结果

cout << cnt << endl;

return 0;

}



#include <stdio.h>

#include <stdlib.h>

int main() {

int arr[100][100]; // 定义一个二维数组存储调研区域的发电量

int r, c, n,

power; // 定义变量：r为调研区域的行数，c为列数，n为电站边长，power为最低要求的发电量

// 读取输入参数：区域的行数（r），列数（c），电站的边长（n），以及最低要求的发电量（power）

scanf("%d %d %d %d", &r, &c, &n, &power);

// 读取调研区域每个位置的发电量

for (int i = 0; i < r; ++i) {

for (int j = 0; j < c; ++j) {

scanf("%d", &arr[i][j]);

}

}

int cnt = 0; // 初始化计数器，记录满足条件的区域数

// 遍历调研区域的每个可能的起始点(i, j)

for (int i = 0; i < r - n + 1; ++i) {

for (int j = 0; j < c - n + 1; ++j) {

int tot = 0; // 初始化总发电量累加器

// 计算从(i, j)为左上角起点的nxn正方形区域总发电量

for (int k = 0; k < n; ++k) {

for (int s = 0; s < n; ++s) {

tot += arr[i + k][j + s]; // 累加该区域内每个位置的发电量

}

}

// 如果该区域的总发电量不低于最低要求的发电量，则计数器加1

if (tot >= power) {

cnt++;

}

}

}

// 输出满足条件的区域数

printf("%d\n", cnt);

return 0;

}



const readline = require('readline');

// 创建接口从标准输入读取数据

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

// 存储从输入读取的数据

let lines = [];

// 监听行输入事件

rl.on('line', (input) => {

lines.push(input);

});

// 监听关闭事件

rl.on('close', () => {

// 解析输入的矩阵和其他参数

let [r, c, n, power] = lines[0].split(' ').map(Number);

let arr = [];

for (let i = 1; i <= r; i++) {

arr.push(lines[i].split(' ').map(Number));

}

let cnt = 0;

// 遍历所有可能的电站位置

for (let i = 0; i <= r - n; i++) {

for (let j = 0; j <= c - n; j++) {

let tot = 0;

// 计算电站区域内的总发电量

for (let k = 0; k < n; k++) {

for (let s = 0; s < n; s++) {

tot += arr[i + k][j + s];

}

}

// 判断是否满足最低发电量要求

if (tot >= power) {

cnt++;

}

}

}

// 输出满足条件的区域数量

console.log(cnt);

});



package main

import (

"bufio"

"fmt"

"os"

"strconv"

"strings"

)

func main() {

reader := bufio.NewReader(os.Stdin)

// 读取并解析第一行

firstLine, \_ := reader.ReadString('\n')

firstLineParts := strings.Fields(firstLine)

r, \_ := strconv.Atoi(firstLineParts[0])

c, \_ := strconv.Atoi(firstLineParts[1])

n, \_ := strconv.Atoi(firstLineParts[2])

power, \_ := strconv.Atoi(firstLineParts[3])

// 创建二维数组来存储发电量

arr := make([][]int, r)

for i := 0; i < r; i++ {

arr[i] = make([]int, c)

line, \_ := reader.ReadString('\n')

lineParts := strings.Fields(line)

for j := 0; j < c; j++ {

arr[i][j], \_ = strconv.Atoi(lineParts[j])

}

}

cnt := 0

// 遍历所有可能的电站位置

for i := 0; i <= r-n; i++ {

for j := 0; j <= c-n; j++ {

tot := 0

// 计算电站区域内的总发电量

for k := 0; k < n; k++ {

for s := 0; s < n; s++ {

tot += arr[i+k][j+s]

}

}

// 判断是否满足最低发电量要求

if tot >= power {

cnt++

}

}

}

// 输出满足条件的区域数量

fmt.Println(cnt)

}