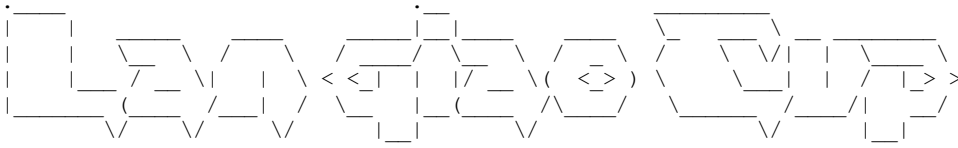


## 蓝桥杯2023



语言 [python3.7](#) 运行终端 [Windows](#) 开源协议 [MIT协议](#)

刷题链接: [https://www.lanqiao.cn/cup/?sort=students\\_count&second\\_category\\_id=3](https://www.lanqiao.cn/cup/?sort=students_count&second_category_id=3)

### 笔记列表:

[505.数字三角形](#)

[497.成绩分析](#)

[598.排序](#)

[594.蛇形填数](#)

[1445.空间](#)

省赛时间: 2023-04-08(周六)

本项目Git Https链接:

<https://github.com/yaunsine/LanqiaoCup.git>

本项目Git SSH链接:

<git@github.com:yaunsine/LanqiaoCup.git>

### 加载环境:

```
pip install -r requirements.txt
```

### markdown文件转pdf方法:

```
python main.py
```

## 1445.空间

本题为填空题，只需要算出结果后，在代码中使用输出语句将所填结果输出即可。

小蓝准备用 256MB 的内存空间开一个数组，数组的每个元素都是 32 位 二进制整数，如果不考虑程序占用的空间和维护内存需要的辅助空间，请问 256MB 的空间可以存储多少个 32 位二进制整数？

```
import os
import sys

# 请在此输入您的代码
print(256 * 1024 * 1024 * 8 // 32)
```

## 497.成绩分析

题目描述 小蓝给学生们组织了一场考试，卷面总分为 100 分，每个学生的得分都是一个 0 到 100 的整数。

请计算这次考试的最高分、最低分和平均分。

输入描述 输入的第一行包含一个整数  $n$  ( $1 \leq n \leq 104$ )，表示考试人数。

接下来  $n$  行，每行包含一个 0 至 100 的整数，表示一个学生的得分。

输出描述 输出三行。

第一行包含一个整数，表示最高分。

第二行包含一个整数，表示最低分。

第三行包含一个实数，四舍五入保留正好两位小数，表示平均分。

输入输出样例 示例 输入

7 80 92 56 74 88 99 10 copy 输出

99 10 71.29

```
import os
import sys

# 请在此输入您的代码
n = int(input()) # 考试人数
sum_score = 0 # 总分
max_score = 0 # 最高分
min_score = 0 # 最低分
for i in range(n):
    score = int(input()) # 考试得分
    if i == 0:
        min_score = score
    if min_score > score:
        min_score = score
    if max_score < score:
        max_score = score
    sum_score += score

print(max_score)
print(min_score)
print("%.2f"%(sum_score / n))
```

此题注意细节

## 505数字三角形问题



上图给出了一个数字三角形。从三角形的顶部到底部有很多条不同的路径。对于每条路径，把路径上面的数加起来可以得到一个和，你的任务就是找到最大的和。

路径上的每一步只能从一个数走到下一层和它最近的左边的那个数或者右 边的那个数。此外，向左下走的次数与向右下走的次数相差不能超过 1。

输入描述 输入的第一行包含一个整数 N

( 1 ≤ N ≤ 100 ) N (1≤N≤100)，表示三角形的行数。

下面的 N 行给出数字三角形。数字三角形上的数都是 0 至 100 之间的整数。

输出描述 输出一个整数，表示答案。

输入输出样例 示例 输入

```
5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

输出

代码实现：

```
import os
import sys

# 请在此输入您的代码
n = int(input())
i = 1
all_ls = list()
while i <= n:
    line = input()
    row_ls = line.split()
    row_ls = list(map(int, row_ls))
    all_ls.append(row_ls)
    i += 1

for i in range(1, n):
    for j in range(0, i + 1):
        if j == 0:
            all_ls[i][j] += all_ls[i - 1][j]
        elif j == i:
            all_ls[i][j] += all_ls[i - 1][j - 1]
        else:
            all_ls[i][j] += max(all_ls[i - 1][j - 1], all_ls[i - 1][j])

print(max(all_ls[-1][n // 2 - 1], all_ls[-1][n // 2]))
```

## 598.排序

**题目描述** 本题为填空题，只需要算出结果后，在代码中使用输出语句将所填结果输出即可。

小蓝最近学习了一些排序算法，其中冒泡排序让他印象深刻。

在冒泡排序中，每次只能交换相邻的两个元素。

小蓝发现，如果对一个字符串中的字符排序，只允许交换相邻的两个字符，则在所有可能的排序方案中，冒泡排序的总交换次数是最少的。

例如，对于字符串 `lan` 排序，只需要 1 1 次交换。对于字符串 `qiao` 排序，总共需要 4 4 次交换。

小蓝找到了很多字符串试图排序，他恰巧碰到一个字符串，需要 100 100 次交换，可是他忘了吧这个字符串记下来，现在找不到了。

请帮助小蓝找一个只包含小写英文字母且没有字母重复出现的字符串，对该串的字符排序，正好需要 100 100 次交换。如果可能找到多个，请告诉小蓝最短的那个。如果最短的仍然有多个，请告诉小蓝字典序最小的那个。

```
import os
import sys

# 请在此输入您的代码
sum_x = 0
i = 1
for i in range(1, 101):
    sum_x += i
    if sum_x >= 100:
        break
x = sum_x - 100
result = str()
sum_x = 0
first_letter = str()
for j in range(i, -1, -1):
    if i - j == x:
        first_letter = str(chr(97+j))
    else:
        result += str(chr(97+j))
print(first_letter + result)
```

## 594.蛇形填数

**题目描述** 本题为填空题，只需要算出结果后，在代码中使用输出语句将所填结果输出即可。

如下图所示，小明用从 1 开始的正整数“蛇形”填充无限大的矩阵。

```
1 2 6 7 15 ...
3 5 8 14 ...
4 9 13 ...
10 12 ...
11 ...
...
```

容易看出矩阵第二行第二列中的数是 5。请你计算矩阵中第 20 行第 20 列的数是多少？

### 【解法1】

#### 模拟动态规划

```
import os
import sys

mat = [[0 for _ in range(200)] for _ in range(200)]

i = 0
j = 0
mat[i][j] = 1
cnt = 1
while mat[19][19] == 0:
    # 右移
    j += 1
    cnt += 1
    mat[i][j] = cnt
    # 左下角
    while j != 0:
        i += 1
        j -= 1
        cnt += 1
        mat[i][j] = mat[i][j]
    # 下移
    i += 1
    cnt += 1
    mat[i][j] = cnt
    # 右上角
    while i != 0:
        i -= 1
        j += 1
        cnt += 1
        mat[i][j] = cnt

print(mat[19][19])
```

### 【解法2】

#### 对角线算法

```
res = 1
for i in range(0, 20, 1):
    res += i * 4
print(res)
```