

---

# 2024 计算机编程素养提升交流活动

## 模拟赛

时间：2024 年 7 月 7 日 08 : 00 - 11: 30

### 一. 题目概况

题目名称	矩形计数	兜圈子	扩大GCD	SOS
可执行文件名	rectangle.exe	loop.exe	gcd.exe	sos.exe
输入文件名	rectangle.in	loop.in	gcd.in	sos.in
输出文件名	rectangle.out	loop.out	gcd.out	sos.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
题目类型	传统	传统	传统	传统
运行内存上限	256M	256M	256M	256M

### 二. 提交源程序文件名

C++语言	rectangle.cpp	loop.cpp	gcd.cpp	sos.cpp
-------	---------------	----------	---------	---------

### 三. 编译命令

C++语言	-O2 -std=c++14 -Wl,--stack=536870912
-------	--------------------------------------

---

## 矩形计数(rectangle)

### 【问题描述】

在一个平面直角坐标系上有  $N$  个不同的点，点  $i$  ( $1 \leq i \leq N$ ) 的坐标是  $(x_i, y_i)$ 。

请你数一下有多少个矩形，满足：

- 矩形的 4 个顶点都在给定的  $N$  个点中；
- 所有的边都平行于坐标轴。

### 【输入格式】

输入文件名为 `rectangle.in`。

第一行为  $N$ ，接下来  $N$  行，每行为  $x_i y_i$ 。

### 【输出格式】

输出文件名为 `rectangle.out`。

输出文件只有一个整数，表示答案。

### 【输入输出样例】

<code>rectangle.in</code>	<code>rectangle.out</code>
6 0 0 0 1 1 0 1 1 2 0 2 1	3
4 0 1 1 2 2 3 3 4	0
7 0 1 1 0 2 0 2 1 2 2 3 0 3 2	1

---

**【数据规模与约定】**

对于 30% 的数据， $4 \leq N \leq 50$ ， $0 \leq x_i, y_i \leq 2000$ 。

对于 60% 的数据， $4 \leq N \leq 2000$ ， $0 \leq x_i, y_i \leq 2000$ 。

对于 100% 的数据， $4 \leq N \leq 2000$ ， $0 \leq x_i, y_i \leq 10^9$ 。

## 兜圈子(loop)

### 【问题描述】

张三所在的城市里有  $N$  个景点  $M$  条单行道。

第  $i$  条单行道从  $A_i$  到  $B_i$ ，耗时为  $C_i$ 。可能会有重边和自环。

对于每一个景点  $i$ ，问能否从  $i$  出发兜个圈子再回到该景点？

如果可以从  $i$  出发返回到景点  $i$ ，输出花费的最短时间，否则输出  $-1$ 。

### 【输入格式】

输入文件名为 `loop.in`。

第一行为  $N\ M$ ，接下来  $M$  行，每行为  $A_i\ B_i\ C_i$ 。

### 【输出格式】

输出文件名为 `loop.out`。

输出  $N$  行，第  $i$  行为从点  $i$  出发回到点  $i$  的最短时间，如果不能回到原点，输出  $-1$ 。

### 【输入输出样例】

loop.in	loop.out
4 4 1 2 5 2 3 10 3 1 15 4 3 20	30 30 30 -1
4 6 1 2 5 1 3 10 2 4 5 3 4 10 4 1 10 1 1 10	10 20 30 20
4 7 1 2 10 2 3 30 1 4 15 3 4 25 3 4 20 4 3 20 4 3 30	-1 -1 40 40

---

**【数据规模与约定】**

对于 30% 的数据， $1 \leq N, M \leq 300$ 。

对于 100% 的数据， $1 \leq N, M \leq 2000$ ， $1 \leq A_i \leq N$ ， $1 \leq B_i \leq N$ ， $0 \leq C_i \leq 100000$ 。

---

## 扩大GCD(gcd)

### 【问题描述】

给定  $n$  个数  $a_1, a_2, \dots, a_n$ ，请你去掉尽量少的数，使得剩下数的最大公约数比原来的大。

输出去掉数的个数，若不存在使GCD变大的方案，则输出 -1。

### 【输入格式】

输入文件名为 gcd.in。

第一行为  $n$ ，接下来一行，为  $a_1, a_2, \dots, a_n$ ，用空格隔开。

### 【输出格式】

输出文件名为 gcd.out。

输出文件只有一个整数，表示答案。

### 【输入输出样例】

gcd.in	gcd.out
3 1 2 4	1
4 6 9 15 30	2
3 1 1 1	-1

### 【数据规模与约定】

对于 30% 的数据， $2 \leq N \leq 20$ ， $1 \leq a_i \leq 10000$ 。

对于 60% 的数据， $2 \leq N \leq 2000$ ， $1 \leq a_i \leq 1000000$ 。

对于 100% 的数据， $2 \leq N \leq 300000$ ， $1 \leq a_i \leq 15000000$ 。

---

## SOS(sos)

### 【问题描述】

给定一个  $1 \times N$  的空白棋盘，Alice 和 Bob 轮流在空格里填字，只可以填 S 或 O，不允许跳过填字。

当一方填完一个字后，若出现连续的三个格子正好是 SOS，则刚填字的一方就获胜。若格子填满而没有人获胜，则两方打平。

给定 N，问在双方最优策略情况下，先手必胜、必败还是必平？

### 【输入格式】

输入文件名为 `sos.in`。

多组测试数据，第一行为数据组数 T。

接下来 T 行，每行输入一个正整数 N。

### 【输出格式】

输出文件名为 `sos.out`。

输出 T 行答案。先手胜为1，负为-1，平为0。

### 【输入输出样例】

<code>sos.in</code>	<code>sos.out</code>
3	0
3	1
7	-1
18	

### 【数据规模与约定】

对于 30% 的数据， $1 \leq N \leq 20$ 。

对于 100% 的数据， $1 \leq T \leq 10, 1 \leq N \leq 1000000000$ 。