

# 苏州市人工智能学会

## 青少年人工智能核心算法素养考核(SACCC)

### 五级

时间：2026 年 2 月 7 日 08:30 ~ 12:00

题目名称	紧挨	家庭	选材	握手
题目类型	传统型	传统型	传统型	传统型
目录	close	family	select	hand
可执行文件名	close	family	select	hand
输入文件名	close.in	family.in	select.in	hand.in
输出文件名	close.out	family.out	select.out	hand.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	10	10	10	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	close.cpp	family.cpp	select.cpp	hand.cpp
-----------	-----------	------------	------------	----------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

**注意事项（请仔细阅读）**

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. `main` 函数的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考监考老师的具体要求。
4. 因违反以上三点而出现的错误或问题，产生的后果由本人承担。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。

## 紧挨 (close)

### 【题目背景】

SACCC 苏州市人工智能核心算法能力认证，英文名 Suzhou AI Core Algorithm Competence Certification (以下简称 SACCC)，由苏州市人工智能学会发起并主办，是为计算机和编程学习者提供学业能力验证的平台。SACCC 旨在提升苏州市青少年人工智能核心算法能力、在中小学推广和普及人工智能核心算法学习和编程教育。

**子序列：**是指在序列中选择某些元素，按照序列中的顺序连接在一起形成的。比如 SAC、SC、CC、SCC 都是 SACCC 的子序列。但 CSA 这个序列不是。我们定义子序列长度为子序列中包含的元素数量，比如 SAC 的长度为 3，CC 的长度为 2。如果没有特殊说明，空序列和整个序列都是原序列的一个子序列。我们一般会用非空子序列来排除空序列，用真子序列来排除整个序列，用非空真子序列来同时排除两个。

**子串：**是指在序列中选择连续的某些元素，按照序列中的顺序连接在一起形成的，所以子串也常被称作连续子序列。比如 SAC、CCC 都是 SACCC 的子串。但 SC 不是。与子序列相同，子串包含的元素个数也被称为子串的长度，同时也有“非空”、“真”等描述方法。

### 【题目描述】

现在 小爱同学 得到了一个仅包含英文大写字母的字符串  $s$ ，小爱同学 想要在其中找到一个子串  $t$ ，使得 SACCC 是  $t$  的子序列。显然有可能有多个这样的  $t$ ，小爱同学 想要一个知道最短的  $t$  的长度是多少。

比如当  $s$  为 SSSDEFGHIJKABCBCSCCASCSCCCCC 时，子串 SDEFGHIJKABCBCSCCASCSCC、SDEFGHIJKABCBC、SCCASCSCC 等都包含子序列 SACCC。其中最短的是 SCCASCSCC，长度为 8。

```
1 SSSDEFGHIJKABCBCSCCASCSCCCCC
2           *   *   ***
3           ^^^^^^^^
```

### 【输入格式】

输入一行，即字符串  $s$ 。

### 【输出格式】

一行一个整数，表示  $t$  最小长度。

### 【样例 1 输入】

```
1 SSSDEFGHIJKABCBCSCCASCSCCCCC
```

**【样例 1 输出】**

1

8

**【样例 2 输入】**

1

DDDSACCTCE

**【样例 2 输出】**

1

6

**【数据规模与约定】**

对于 100% 的数据满足： $s$  中仅包含大写英文字母且包含子序列 SACCC， $1 \leq |s| \leq 1000$ 。其中  $|s|$  表示字符串  $s$  的长度。

- 对于其中 30% 的数据满足： $|s| \leq 20$
- 对于另 30% 的数据满足： $s$  中 S、A 两个字符都只出现一次，C 只出现了三次。换言之， $s$  中仅有一个子序列是 SACCC。

## 家庭 (family)

### 【题目描述】

近几十年来，每一位朋友来到这个世界的时候，人工智能(AI)的研究和发展都在默默前行。比如一个家庭中父亲出生于1970年3月10日，AI低谷期；母亲出生于1993年5月20日，AI复苏期；孩子出生于2016年3月9日，AlphaGo战胜李世石前夕(AI应用于围棋)。

父亲在AI寒冬中坚持研究，母亲在复苏期利用AI工具创业。孩子出生时，AI正迈向新高度。家庭日记记录关键事件：父亲在低谷期开发专家系统，母亲在繁荣期创建AI教育平台，孩子在成长中体验智能家居。每到出生纪念日到来，全家都聚在一起回顾AI如何伴随人生重要时刻，从挫折到辉煌。

当我们想到自己出生的年月日的时候，总感觉这是一串神秘的数字，伴随我们终身不忘；而一家三口（父亲、母亲、一个孩子）的出生日期放在一起，又会有什么奇妙的事情发生呢？现在请你按照要求得到一个数值或者判断其是否合法。

会有多组询问，每组询问会给出三个人生日的年月日  $y m d$  和一个用于取模的整数  $MOD$ 。

- 1、判断年月日合法：每一组生日的年月日是否真实存在，如果不合法则直接输出“FALSE”。注意：不需要判断孩子生日和父母生日的大小关系。
- 2、如果合法，请计算出每个人从出生到今天（2026年2月7日）的实际天数  $AGE$ ，比如某人是2025年2月6日出生的，则  $AGE$  为366；再根据  $AGE$  计算出秒数  $SEC$ ，而  $SEC$  为  $AGE \times 24 \times 3600$ ；继而将父亲的秒数  $SECfa$ ，乘以母亲的秒数  $SECma$ ，再除以  $MOD$  得到一个余数  $mod1$ ；将孩子的秒数  $SECchild$  除以  $MOD$  得到另一个余数  $mod2$ ；请你输出  $mod1$  与  $mod2$  之间的合数之和（该统计的区间包括端点  $mod1$  和  $mod2$ ）。

此处合数的含义是大于2且至少含有3个约数的整数，比如6有1、2、3、6这4个约数，比如4有1、2、4这3个约数，所以6和4都是合数，但是比如7只有1、7这2个约数，所以7不是合数。

### 【输入格式】

第一行是一个整数  $n$ ，表示询问次数。

接下来是  $4*n$  行，每组询问占4行，

前三行每一行依次表示父亲、母亲、孩子的出生日期，每行3个整数  $y m d$ ，两两之间用一个空格隔开，表示年、月、日；

第四行是一个整数，表示一个用于取模的整数  $MOD$ 。

### 【输出格式】

共  $n$  行，每行要么是“FALSE”，不含双引号，要么是一个整数。

**【样例 1 输入】**

```
1 1
2 1999 12 31
3 2000 1 2
4 2025 9 30
5 23
```

**【样例 1 输出】**

```
1 102
```

**【样例 1 说明】**

第一组询问中，父亲的年龄总天数是 9535，母亲的年龄总天数是 9533，孩子的年龄总天数是 130， $mod1$  为 7， $mod2$  为 19，此处  $mod1$  较小， $mod2$  较大， $[7, 19]$  区间之内的合数之和为 102，这些合数分别为 8、9、10、12、14、15、16、18。

**【样例 2 输入】**

```
1 2
2 1980 2 30
3 1981 1 4
4 2005 10 9
5 1234567
6 1980 2 28
7 1981 1 4
8 2005 10 9
9 10000000
```

**【样例 2 输出】**

```
1 FALSE
2 23026357624828
```

**【样例 2 说明】**

第一组询问中，父亲的出生年月日中显示是 2 月份，虽然 1980 年是闰年，2 月份最多 29 天，但是显示 2 月 30 日，是不合法的。

第二组询问中, 父亲的年龄总天数是 16781, 母亲的年龄总天数是 16470, 孩子的年龄总天数是 7426,  $mod1$  为 7200000,  $mod2$  为 1606400, 此处  $mod2$  较小,  $mod1$  较大,  $[1606400, 7200000]$  区间之内的合数之和为 23026357624828。

### 【样例 3】

见选手目录下的 *family/family3.in* 与 *family/family3.ans*。

### 【数据范围】

所有输入的  $y, m, d$  肯定满足  $1900 \leq y \leq 2025, 1 \leq m \leq 12, 1 \leq d \leq 31$

- 对于 40% 的数据满足:  $n = 1, 1 \leq MOD \leq 100$
- 对于 70% 的数据满足:  $1 \leq n \leq 5000, 1 \leq MOD \leq 5000$
- 对于 100% 的数据满足:  $1 \leq n \leq 5 \times 10^4, 1 \leq MOD \leq 10^7$

## 选材 (select)

### 【题目描述】

在人工智能 (AI) 的优化算法领域, 我们面临一个经典的选择问题: 给定一个由  $n$  个智能个体组成的集合, 每个个体具有特定的智商值 (IQ), 我们的目标是通过智能筛选机制, 从  $n$  个个体中选出  $\lfloor n/2 \rfloor$  个成员, 使得所选群体的 IQ 总和  $X$  与未选群体的 IQ 总和  $Y$  之间差的绝对值最小化。

其中的  $\lfloor n/2 \rfloor$  表示对  $n/2$  向下取整, 即相当于得到的商舍去小数部分。

这一挑战本质上属于组合优化问题, 旨在通过高效算法实现资源的最优分配, 例如在团队组建或任务分配场景中平衡能力差异。

现在请你根据给定数据找到某种特定的选择方案, 满足  $X$  与  $Y$  之间差的绝对值最小, 输出这个最小值。

### 【输入格式】

第一行包含一个正整数  $n$ , 表示智能个体的数量。

第二行包含  $n$  个用空格隔开的正整数, 表示每个智能个体的 IQ 值  $a_i$ 。

### 【输出格式】

一个整数, 就是某种特定选择方案下得到的  $X$  与  $Y$  之间差的最小绝对值。

### 【样例 1 输入】

```
1 6
2 5 3 3 3 3 4
```

### 【样例 1 输出】

```
1 1
```

### 【样例 1 解释】

如果选择第 2、3、4 号为所选群体, 其 IQ 总和  $X = 3 + 3 + 3 = 9$ , 未选群体的 IQ 总和  $Y = 5 + 3 + 4 = 12$ , 差的绝对值是 3。

如果选择第 1、2、3 号为所选群体, 其 IQ 总和  $X = 5 + 3 + 3 = 11$ , 未选群体的 IQ 总和  $Y = 3 + 3 + 4 = 10$ , 差的绝对值是 1。

找不到比 1 更小的方案了。

**【样例 2 输入】**

```
1 5
2 8 10 10 10 7
```

**【样例 2 输出】**

```
1 5
```

**【样例 2 解释】**

选择第 2、3 号为所选群体。

**【样例 3】**

见选手目录下的 *select/select3.in* 与 *select/select3.ans*。

**【数据范围】**

- 对于 40% 的数据满足：  $2 \leq n \leq 6, 1 \leq a_i \leq 10$
- 对于 70% 的数据满足：  $2 \leq n \leq 24, 1 \leq a_i \leq 500$
- 对于 100% 的数据满足：  $2 \leq n \leq 36, 1 \leq a_i \leq 10^{15}$

## 握手 (hand)

### 【题目描述】

现在很多人都在使用人工智能(AI)辅助工作和生活,但 AI 的使用往往离不开网络技术的支撑,用户提出的诉求需要通过网络传输到远程服务器进行分析然后给出相应的结果或者帮助;而在网络技术里的“握手”(Handshake)是个核心概念,它指的是通信双方在正式传输数据前,通过一系列消息交换来协商参数、建立连接的过程;而不同的网络协议对于“握手”的次数要求也不尽相同。

生活中,人与人之间的握手也是非常频繁的,而人们一般都只与熟悉的并且关系好的朋友握手,当然朋友可以没有,也可以有多个。

现在 A 公司的员工到另外一个兄弟单位 B 公司交流学习,那么这两个公司员工之间免不了有相互熟悉的朋友进行握手。注意:这里可能进行握手的朋友关系仅限于不同公司之间。

假设 A 公司的员工排成第一列, B 公司的员工排成第二列,两列人员面对面平行站立。

据统计,两个公司员工之间存在  $n$  对朋友关系,编号从 1 到  $n$ ,即站在第一列坐标  $A_i$  位置上的 A 公司员工和站在第二列坐标  $B_i$  的位置上的 B 公司员工之间存在朋友关系。

每个人都想尽快和老朋友相聚到各自所在位置连线的中点进行握手,而所有握手的过程是同时进行的,为了避免产生人员碰撞的安全事故,所有进行握手的朋友之间的连线不能有交点。

注意:一个人可能有多个朋友,但最多只能和一个朋友握手。

在遵守上述限制条件的前提下,最多有多少对朋友可以同时握手?

### 【输入格式】

输入数据都按以下格式给出:

```
n
A1 B1
A2 B2
⋮
An Bn
```

### 【输出格式】

一个整数,表示最多可以同时握手数量。

### 【样例 1 输入】

```
1 3
2 3 5
```

```
3 1 4
4 2 6
```

**【样例 1 输出】**

```
1 2
```

**【样例 1 解释】**

第 1 对、第 2 对朋友可以同时握手，但第 1 对、第 3 对朋友无法同时握手，因为“**A** 公司第一列的 3 位置到 **B** 公司第二列的 5 位置之间的连线”与“**A** 公司第一列的 2 位置到 **B** 公司第二列的 6 位置之间的连线”两者有交点。

因此，只要选择合适的朋友关系，最多可以让 2 对朋友同时握手，而 3 对同时握手是不安全的。故答案为 2。

**【样例 2 输入】**

```
1 5
2 1 2
3 1 3
4 1 4
5 1 5
6 1 6
```

**【样例 2 输出】**

```
1 1
```

**【样例 3】**

见选手目录下的 *hand/hand3.in* 与 *hand/hand3.ans*。

**【数据范围】**

- 对于 10% 的数据满足：  $n = 2$
- 对于 40% 的数据满足：  $n \leq 20$
- 对于 50% 的数据满足： 出现的所有  $A_i$  互不相同，所有的  $B_i$  互不相同
- 对于 70% 的数据满足：  $n \leq 5000$
- 对于 100% 的数据满足：  $1 \leq n \leq 2 \times 10^5$ ，  $0 \leq A_i \leq 10^9$ ，  $0 \leq B_i \leq 10^9$
- 输入的所有数值均为整数。