

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）  
4 级

时间：2026 年 2 月 7 日 08:30 ~ 2027 年 2 月 7 日 11:10

题目名称	算力	数字	数独	付费	分割	硬币
题目类型	传统型	传统型	传统型	传统型	传统型	传统型
每个测试点时限	1.0 秒					
内存限制	256 MiB	256 MiB	256 MiB	256 MiB	512 MiB	512 MiB
子任务数目	5	5	10	10	10	10

编译选项

对于 C++ 语言	<code>-std=c++14 -O2 -static</code>
-----------	-------------------------------------

## 算力 (power)

### 【题目描述】

小智上网搜索到了一份资料，其中记录了  $n$  个不同的大语言模型。大语言模型的大小对其性能至关重要，通常用模型中的参数数量表示，如 1B、32B、70B 等。在这份资料中，第  $i$  个模型的参数数量为  $a_i$ 。

小智希望选择合适的大语言模型：参数数量不能太少——确保人工智能具有一定能力；参数数量也不能太大——他的计算机并没有强大的算力。他研究并确定了两个参数  $l$  和  $r$ ，给定所有大语言模型的参数大小，请你编写一个程序，帮他统计有多少个模型的大小恰好在  $l$  和  $r$  之间（包含两端）。

### 【输入格式】

从文件 `power.in` 中读入数据。

第一行一个整数  $n$ 。

第二行  $n$  个整数，表示  $a_i$  的值。

第三行两个整数  $l$ 、 $r$ 。

### 【输出格式】

输出到文件 `power.out` 中。

输出一行一个整数表示答案。

### 【样例 1 输入】

```
1 3
2 100 150 200
3 120 300
```

### 【样例 1 输出】

```
1 2
```

### 【样例 2 输入】

```
1 5
2 110 120 130 140 150
3 500 1000
```

**【样例 2 输出】**

```
1 0
```

**【样例 3 输入】**

```
1 5
2 200 300 50 40 20
3 1 999
```

**【样例 3 输出】**

```
1 5
```

**【数据范围】**

- 对于 40% 的数据，保证  $h_i \leq 1000$ 。
- 对于全部测试数据，保证： $1 \leq n \leq 2 \times 10^5$ ， $1 \leq h_i \leq 10^9$ ， $1 \leq l \leq r \leq 10^9$ 。

## 数字 (digits)

### 【题目描述】

给你一个十进制整数，请你倒序输出整数的每一位数字。

### 【输入格式】

从文件 `digits.in` 中读入数据。

输入共一行一个整数。

### 【输出格式】

输出到文件 `digits.out` 中。

输出一行，倒序输出这个整数的每一位。如果倒序输出的结果中有前导零，**不要**舍去这些字符。

### 【样例 1 输入】

1 5678

### 【样例 1 输出】

1 8765

### 【样例 2 输入】

1 43210

### 【样例 2 输出】

1 01234

### 【样例 3 输入】

1 65536

**【样例 3 输出】**

1 63556

**【数据范围】**

- 对于 40% 的数据，保证输入的数字不超过 1000。
- 对于全部测试数据，保证输入的数字是正整数，且不超过  $2 \times 10^9$ 。

## 数独 (sudoku)

### 【题目描述】

小智很喜欢玩数独。在数独游戏中，玩家需要在一个  $9 \times 9$  的方格内填上 1 到 9 之间的数字。这个  $9 \times 9$  方格被粗线分成 9 个  $3 \times 3$  的宫格。玩家填入的数字需要同时满足下列三个条件：

- 行：每一横行（共 9 行）的数字必须是 1 ~ 9，不能有重复。
- 列：每一竖列（共 9 列）的数字必须是 1 ~ 9，不能有重复。
- 宫：每一个  $3 \times 3$  的粗线宫格内的数字必须是 1 ~ 9，不能有重复。

例如，下面的第一张图表示初始的数独方格，第二张图表示正确解谜的数独方格，这个结果同时满足上面的三个条件。

3	6	1		5				7
	9		7	1	3	2	6	8
7	2		6		4	3	5	1
1	3	7	5		9	6		4
	8		3	6	7		1	2
2	5	6				9	7	
	4	3	1	7	5	8	2	
	7						3	
		2	9	3	6	7	4	5

3	6	1	2	5	8	4	9	7
4	9	5	7	1	3	2	6	8
7	2	8	6	9	4	3	5	1
1	3	7	5	2	9	6	8	4
9	8	4	3	6	7	5	1	2
2	5	6	4	8	1	9	7	3
6	4	3	1	7	5	8	2	9
5	7	9	8	4	2	1	3	6
8	1	2	9	3	6	7	4	5

小智已经填好了一份数独方格，请你帮他判断他填入的数字是否同时满足上面的三个条件。

### 【输入格式】

从文件 `sudoku.in` 中读入数据。

输入共 9 行，每行一个长度为 9 且仅包含字符 1~9 的字符串。

### 【输出格式】

输出到文件 `sudoku.out` 中。

输出一行一个字符串表示答案：如果小智填入的数字同时符合三个条件，输出 Yes，否则输出 No。

**【样例 1 输入】**

```
1 361258497
2 495713268
3 728694351
4 137529684
5 984367512
6 256481973
7 643175829
8 579842136
9 812936745
```

**【样例 1 输出】**

```
1 Yes
```

**【样例 2 输入】**

```
1 361258497
2 495713268
3 728694351
4 137529684
5 436751298
6 256481973
7 643175829
8 579842136
9 812936745
```

**【样例 2 输出】**

```
1 No
```

**【样例 3 输入】**

```
1 596134872
2 734285619
```

```
3 821796534
4 365412987
5 149867253
6 287953461
7 618349725
8 973528146
9 452671398
```

**【样例 3 输出】**

```
1 Yes
```

**【数据范围】**

对于全部测试数据，保证： $9 \times 9$  方格内的每一个数字都是 1 到 9 之间的整数。

## 付费 (pay)

### 【题目描述】

小智要用人工智能解决一个复杂的问题，经过他的测算，解决这个问题需要消耗他  $n$  万个 token。他必须先人工智能平台上购买足够的 token 额度才能开始使用。

人工智能平台的商店中有三种购买 token 额度的套餐：小套餐、中套餐、大套餐。

- 购买小套餐将获得 3 万个 token 额度，价格为  $a$  元；
- 购买中套餐将获得 5 万个 token 额度，价格为  $b$  元；
- 购买大套餐将获得 10 万个 token 额度，价格为  $c$  元；

小智想知道，他至少要花多少钱，才能买到足够的 token 额度并让人工智能解决问题？请注意，小智购买的 token 额度可以比  $n$  万个更多。

### 【输入格式】

从文件 `pay.in` 中读入数据。

第一行一个整数  $n$ 。

第二行三个整数  $a$ 、 $b$ 、 $c$ 。

### 【输出格式】

输出到文件 `pay.out` 中。

输出一行一个整数表示答案。

### 【样例 1 输入】

```
1 20
2 5 8 13
```

### 【样例 1 输出】

```
1 26
```

### 【样例 1 解释】

小智购买两次大套餐最划算，总共花费  $2 \times 13 = 26$  元。

### 【样例 2 输入】

```
1 53
2 6 8 19
```

**【样例 2 输出】**

```
1 86
```

**【样例 2 解释】**

小智购买一次小套餐和十次中套餐最划算，总共花费  $1 \times 6 + 10 \times 8 = 80$  元。

**【样例 3 输入】**

```
1 99
2 5 7 12
```

**【样例 3 输出】**

```
1 120
```

**【样例 3 解释】**

小智购买十次大套餐最划算，总共花费  $10 \times 12 = 120$  元。尽管小智多买了 1 万个 token 额度，但这样的总开销是最低的。

**【数据范围】**

- 对于 30% 的数据，保证： $n \leq 20$ 。
- 对于 60% 的数据，保证： $n \leq 100$ 。
- 对于全部测试数据，保证： $1 \leq n \leq 2000$ ， $1 \leq a, b, c \leq 100$ 。

## 分割 (divide)

### 【题目描述】

给定一个整数  $x$ ，你可以执行下面三种操作任意多次：

- 如果  $x$  是 2 的倍数，则将  $x$  替换为  $\frac{x}{2}$ ；
- 如果  $x$  是 3 的倍数，则将  $x$  替换为  $\frac{2x}{3}$ ；
- 如果  $x$  是 5 的倍数，则将  $x$  替换为  $\frac{4x}{5}$ 。

请你编写一个程序，计算将  $x$  变成 1 所需要的最小操作次数。

### 【输入格式】

从文件 `divide.in` 中读入数据。

本题有多组测试数据。

第一行一个正整数  $T$ ，表示测试数据的组数。

每组测试数据仅包含一行一个整数  $x$ 。

### 【输出格式】

输出到文件 `divide.out` 中。

对于每组测试数据，输出一行一个整数表示答案。

如果不可能将  $x$  变成 1，则输出 `-1`。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
1 0
2 1
3 3
```

### 【样例 2 输入】

```
1 5
2 2
3 3
4 5
5 7
6 11
```

**【样例 2 输出】**

```
1 1
2 2
3 3
4 -1
5 -1
```

**【样例 3】**

见选手目录下的 `divide/divide3.in` 与 `divide/divide3.ans`。

**【数据范围】**

- 对于 30% 的测试数据，保证： $1 \leq x \leq 1000$ ；
- 对于 60% 的测试数据，保证： $1 \leq x \leq 10^9$ ；
- 对于全部测试数据，保证： $1 \leq T \leq 1000$ ， $1 \leq x \leq 10^{18}$ 。

## 硬币 (coins)

### 【题目描述】

小智收到了  $n$  个硬币，他将这些硬币进行编号，分别用 1 到  $n$  之间的整数表示。

这些硬币中有一些是假币，小智想找出哪些硬币是假币，于是他借来了一台判断硬币真假的机器。小智可以往机器中放入任意数量的硬币，机器将会检测放入的硬币是否存在假币。遗憾的是，这台机器不太灵敏，只有当放入不少于  $k$  个假币的时候，机器才会报警。

小智用这台机器进行了  $m$  次试验。在第  $i$  次试验中，小智进行了如下操作并记录结果：

- 将  $c_i$  枚硬币放入机器，这些硬币的编号分别为  $a_{i,1}, a_{i,2}, \dots, a_{i,c_i}$ 。
- 如果机器报警，也就是放入了不少于  $k$  个假币，则记录结果为  $x$ ，否则记录结果为  $o$ 。

小智知道， $n$  个硬币的真假分布总共有  $2^n$  种不同的情况。他想知道，在这所有的  $2^n$  种情况中，有多少种情况全部符合他记录的  $m$  次试验的结果？请注意，小智记录的结果可能有误，因此可能不存在任意一种情况符合记录结果。

### 【输入格式】

从文件 `coins.in` 中读入数据。

第一行三个整数  $n$ 、 $m$ 、 $k$ 。

接下来  $m$  行，每行数据表示一次试验。每行首先是一个整数  $c_i$ ，随后是  $c_i$  个整数分别表示硬币的编号，最后一个字符  $x$  或  $o$  表示试验的结果。

### 【输出格式】

输出到文件 `coins.out` 中。

输出一行一个整数表示答案。

### 【样例 1 输入】

```
1 2 3 1
2 1 1 o
3 1 2 x
4 2 1 2 x
```

### 【样例 1 输出】

```
1 1
```

**【样例 1 解释】**

小智收到了 2 个硬币，并进行了 3 次试验。

- 根据第一次试验的结果，硬币 1 是真币。
- 根据第二次试验的结果，硬币 2 是假币。
- 根据第三次试验的结果，硬币 1、2 中至少有 1 个假币。

因此，只有一种情况符合全部三次试验的结果：

- 硬币 1 是真币，硬币 2 是假币。

**【样例 2 输入】**

```
1 3 2 2
2 3 1 2 3 o
3 2 1 2 x
```

**【样例 2 输出】**

```
1 0
```

**【样例 2 解释】**

小智收到了 3 个硬币，并进行了 2 次试验。

- 根据第一次试验的结果，硬币 1、2、3 中最多只可能有 1 个假币。
- 根据第二次试验的结果，硬币 1、2 中至少有 2 个假币。

因此，不可能存在符合全部两次试验的假币分布，答案为 0。

**【样例 3 输入】**

```
1 6 4 1
2 1 1 x
3 1 2 x
4 1 3 o
5 1 4 o
```

**【样例 3 输出】**

1 4

**【样例 4 输入】**

```
1 6 4 2
2 3 1 2 3 x
3 4 2 3 4 6 x
4 2 1 3 o
5 3 4 5 6 x
```

**【样例 4 输出】**

1 8

**【数据范围】**

- 对于 10% 的测试数据, 保证:  $k = 1$  且  $c_i = 1$ ;
- 对于 30% 的测试数据, 保证:  $k = 1$ ;
- 对于 30% 的测试数据, 保证:  $n \leq 2$ ;
- 对于全部测试数据, 保证:
  - $1 \leq k \leq n \leq 15$
  - $1 \leq m \leq 100$
  - $1 \leq c_i \leq n$
  - $1 \leq a_{i,j} \leq n$ , 且对于任意的  $i$ ,  $a_{i,j}$  互不相同