

2025 CCF 非专业级软件能力认证

CSP-J/S 2025 第二轮认证

入门级

时间：2025 年 11 月 1 日 08:30 ~ 12:00

题目名称	拼数	座位	异或和	多边形
题目类型	传统型	传统型	传统型	传统型
目录	number	seat	xor	polygon
可执行文件名	number	seat	xor	polygon
输入文件名	number.in	seat.in	xor.in	polygon.in
输出文件名	number.out	seat.out	xor.out	polygon.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	25	20	20	25
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	number.cpp	seat.cpp	xor.cpp	polygon.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. 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换行）。
4. 选手提交的程序源文件大小不得超过 100 KiB。
5. 提交的程序源文件的放置位置请参考各省的具体要求。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 禁止在源代码中改变编译器参数（如使用 `#pragma` 命令），禁止使用系统结构相关指令（如内联汇编）或其他可能造成不公平的方法。
8. 因违反上述规定而出现的问题，申诉时一律不予受理。
9. 只提供 Linux 格式附加样例文件。
10. 全国统一评测时采用的机器配置为：Intel Core Ultra 9 285K CPU @ 3.70 GHz（关闭睿频与能效核），内存 96 GB。上述时限以此配置为准。
11. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

拼数 (number)

【题目描述】

小 R 正在学习字符串处理。小 X 给了小 R 一个字符串 s ，其中 s 仅包含小写英文字母及数字，且包含至少一个 $1 \sim 9$ 中的数字。小 X 希望小 R 使用 s 中的任意多个数字，按任意顺序拼成一个正整数。注意：小 R 可以选择 s 中相同的数字，但每个数字只能使用一次。例如，若 s 为 `1a01b`，则小 R 可以同时选择第 1,3,4 个字符，分别为 1,0,1，拼成正整数 101 或 110；但小 R 不能拼成正整数 111，因为 s 仅包含两个数字 1。

小 R 想知道，在他所有能拼成的正整数中，最大的是多少。你需要帮助小 R 求出他能拼成的正整数的最大值。

【输入格式】

从文件 `number.in` 中读入数据。

输入的第一行包含一个字符串 s ，表示小 X 给小 R 的字符串。

【输出格式】

输出到文件 `number.out` 中。

输出一行一个正整数，表示小 R 能拼成的正整数的最大值。

【样例 1 输入】

1 5

【样例 1 输出】

1 5

【样例 1 解释】

s 仅包含一个数字 5，因此小 R 仅能拼成正整数 5。

【样例 2 输入】

1 290es1q0

【样例 2 输出】

1 92100

【样例 2 解释】

s 包含数字 2,9,0,1,0。可以证明，小 R 拼成的正整数的最大值为 92100。

【样例 3】

见选手目录下的 `number/number3.in` 与 `number/number3.ans`。
该样例满足测试点 9 ~ 11 的约束条件。

【样例 4】

见选手目录下的 `number/number4.in` 与 `number/number4.ans`。
该样例满足测试点 20 的约束条件。

【数据范围】

- 设 $|s|$ 为字符串 s 的长度。对于所有测试数据，保证：
- $1 \leq |s| \leq 10^6$ ；
 - s 仅包含小写英文字母及数字，且包含至少一个 1 ~ 9 中的数字。

测试点编号	$ s \leq$	特殊性质
1	1	A
2	2	
3		无
4	10	A
5, 6		无
7, 8	10^2	A
9 ~ 11		无
12	10^3	A
13, 14		无
15	10^5	A
16, 17		B
18, 19		无
20	10^6	A
21, 22		B
23 ~ 25		无

特殊性质 A: s 仅包含数字。

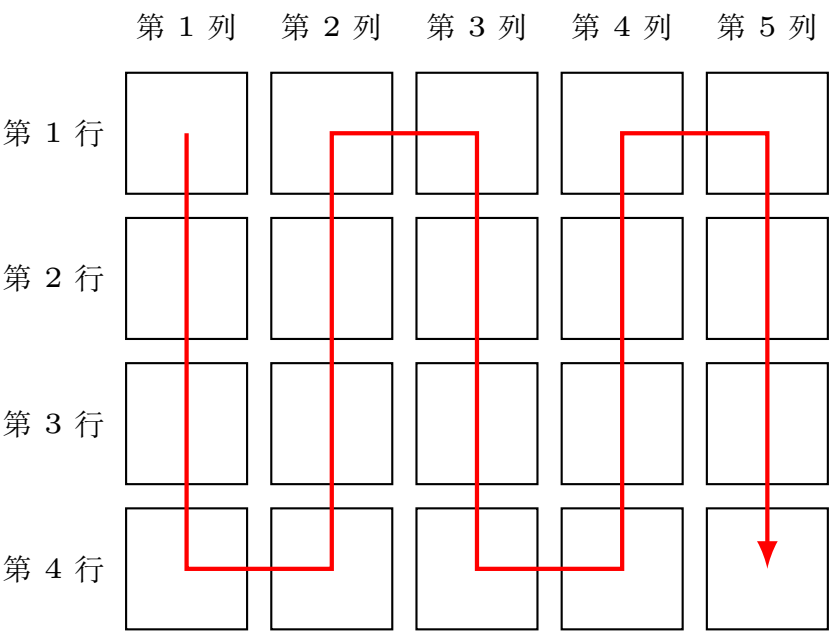
特殊性质 B: s 仅包含不超过 10^3 个数字。

座位 (seat)

【题目描述】

CSP-J 2025 第二轮正在进行。小 R 所在的考场共有 $n \times m$ 名考生，其中所有考生的 CSP-J 2025 第一轮成绩互不相同。所有 $n \times m$ 名考生将按照 CSP-J 2025 第一轮的成绩，由高到低蛇形分配座位，排列成 n 行 m 列。具体地，设小 R 所在的考场的所有考生的成绩从高到低分别为 $s_1 > s_2 > \dots > s_{n \times m}$ ，则成绩为 s_1 的考生的座位为第 1 列第 1 行，成绩为 s_2 的考生的座位为第 1 列第 2 行，……，成绩为 s_n 的考生的座位为第 1 列第 n 行，成绩为 s_{n+1} 的考生的座位为第 2 列第 n 行，……，成绩为 s_{2n} 的考生的座位为第 2 列第 1 行，成绩为 s_{2n+1} 的考生的座位为第 3 列第 1 行，以此类推。

例如，若 $n = 4, m = 5$ ，则所有 $4 \times 5 = 20$ 名考生将按照 CSP-J 2025 第一轮成绩从高到低的顺序，根据下图中的箭头顺序蛇形分配座位。



给定小 R 所在的考场座位的行数 n 与列数 m ，以及小 R 所在的考场的所有考生 CSP-J 2025 第一轮的成绩 $a_1, a_2, \dots, a_{n \times m}$ ，其中 a_1 为小 R CSP-J 2025 第一轮的成绩，你需要帮助小 R 求出，他的座位为第几列第几行。

【输入格式】

从文件 `seat.in` 中读入数据。
输入的第一行包含两个正整数 n, m ，分别表示小 R 所在的考场座位的行数与列数。
输入的第二行包含 $n \times m$ 个正整数 $a_1, a_2, \dots, a_{n \times m}$ ，分别表示小 R 所在的考场的所有考生 CSP-J 2025 第一轮的成绩，其中 a_1 为小 R CSP-J 2025 第一轮的成绩。

【输出格式】

输出到文件 *seat.out* 中。

输出一行两个正整数 c, r ，表示小 R 的座位为第 c 列第 r 行。

【样例 1 输入】

```
1 2 2
2 99 100 97 98
```

【样例 1 输出】

```
1 1 2
```

【样例 1 解释】

按照成绩从高到低的顺序，成绩为 100 的考生的座位为第 1 列第 1 行，成绩为 99 的考生的座位为第 1 列第 2 行，成绩为 98 的考生的座位为第 2 列第 2 行，成绩为 97 的考生的座位为第 2 列第 1 行。小 R 的成绩为 99，因此座位为第 1 列第 2 行。

【样例 2 输入】

```
1 2 2
2 98 99 100 97
```

【样例 2 输出】

```
1 2 2
```

【样例 2 解释】

按照成绩从高到低的顺序，成绩为 100 的考生的座位为第 1 列第 1 行，成绩为 99 的考生的座位为第 1 列第 2 行，成绩为 98 的考生的座位为第 2 列第 2 行，成绩为 97 的考生的座位为第 2 列第 1 行。小 R 的成绩为 98，因此座位为第 2 列第 2 行。

【样例 3 输入】

```
1 3 3
2 94 95 96 97 98 99 100 93 92
```

【样例 3 输出】

```
1 3 1
```

【数据范围】

- 对于所有测试数据，保证：
- $1 \leq n \leq 10, 1 \leq m \leq 10$;
 - 对于所有 $1 \leq i \leq n \times m$ ，均有 $1 \leq a_i \leq 100$ ，且 $a_1, a_2, \dots, a_{n \times m}$ 互不相同。

测试点编号	$n \leq$	$m \leq$	特殊性质
1	1	1	AB
2, 3		10	无
4, 5	10	1	
6	2	2	A
7			B
8, 9			无
10		10	A
11			B
12 ~ 14			无
15 ~ 17	10	2	
18 ~ 20		10	

特殊性质 A：对于所有 $1 \leq i \leq n \times m$ ，均有 $a_i = i$ 。

特殊性质 B：对于所有 $1 \leq i \leq n \times m$ ，均有 $a_i = n \times m - i + 1$ 。

异或和 (xor)

【题目描述】

小 R 有一个长度为 n 的非负整数序列 a_1, a_2, \dots, a_n 。定义一个区间 $[l, r]$ ($1 \leq l \leq r \leq n$) 的权值为 a_l, a_{l+1}, \dots, a_r 的二进制按位异或和, 即 $a_l \oplus a_{l+1} \oplus \dots \oplus a_r$, 其中 \oplus 表示二进制按位异或。

小 X 给了小 R 一个非负整数 k 。小 X 希望小 R 选择序列中尽可能多的不相交的区间, 使得每个区间的权值均为 k 。两个区间 $[l_1, r_1], [l_2, r_2]$ 相交当且仅当两个区间同时包含至少一个相同的下标, 即存在 $1 \leq i \leq n$ 使得 $l_1 \leq i \leq r_1$ 且 $l_2 \leq i \leq r_2$ 。

例如, 对于序列 $[2, 1, 0, 3]$, 若 $k = 2$, 则小 R 可以选择区间 $[1, 1]$ 和区间 $[2, 4]$, 权值分别为 2 和 $1 \oplus 0 \oplus 3 = 2$; 若 $k = 3$, 则小 R 可以选择区间 $[1, 2]$ 和区间 $[4, 4]$, 权值分别为 $1 \oplus 2 = 3$ 和 3。

你需要帮助小 R 求出他能选出的区间数量的最大值。

【输入格式】

从文件 `xor.in` 中读入数据。

输入的第一行包含两个非负整数 n, k , 分别表示小 R 的序列长度和小 X 给小 R 的非负整数。

输入的第二行包含 n 个非负整数 a_1, a_2, \dots, a_n , 表示小 R 的序列。

【输出格式】

输出到文件 `xor.out` 中。

输出一行一个非负整数, 表示小 R 能选出的区间数量的最大值。

【样例 1 输入】

```
1 4 2
2 2 1 0 3
```

【样例 1 输出】

```
1 2
```

【样例 1 解释】

小 R 可以选择区间 $[1, 1]$ 和区间 $[2, 4]$, 异或和分别为 2 和 $1 \oplus 0 \oplus 3 = 2$ 。可以证明, 小 R 能选出的区间数量的最大值为 2。

【样例 2 输入】

```
1 4 3
2 2 1 0 3
```

【样例 2 输出】

```
1 2
```

【样例 2 解释】

小 R 可以选择区间 $[1, 2]$ 和区间 $[4, 4]$ ，异或和分别为 $1 \oplus 2 = 3$ 和 3。可以证明，小 R 能选出的区间数量的最大值为 2。

【样例 3 输入】

```
1 4 0
2 2 1 0 3
```

【样例 3 输出】

```
1 1
```

【样例 3 解释】

小 R 可以选择区间 $[3, 3]$ ，异或和为 0。可以证明，小 R 能选出的区间数量的最大值为 1。注意：小 R 不能同时选择区间 $[3, 3]$ 和区间 $[1, 4]$ ，因为这两个区间同时包含下标 3。

【样例 4】

见选手目录下的 *xor/xor4.in* 与 *xor/xor4.ans*。
该样例满足测试点 4,5 的约束条件。

【样例 5】

见选手目录下的 *xor/xor5.in* 与 *xor/xor5.ans*。
该样例满足测试点 9,10 的约束条件。

【样例 6】

见选手目录下的 *xor/xor6.in* 与 *xor/xor6.ans*。

该样例满足测试点 14, 15 的约束条件。

【数据范围】

对于所有测试数据，保证：

- $1 \leq n \leq 5 \times 10^5$, $0 \leq k < 2^{20}$;
- 对于所有 $1 \leq i \leq n$, 均有 $0 \leq a_i < 2^{20}$ 。

测试点编号	$n \leq$	k	特殊性质
1	2	$= 0$	A
2	10	≤ 1	B
3	10^2	$= 0$	A
4, 5		≤ 1	B
6 ~ 8		≤ 255	C
9, 10	10^3	$< 2^{20}$	无
11, 12			
13	2×10^5	≤ 1	B
14, 15		≤ 255	C
16		$< 2^{20}$	无
17	5×10^5	≤ 255	C
18 ~ 20		$< 2^{20}$	无

特殊性质 A：对于所有 $1 \leq i \leq n$, 均有 $a_i = 1$ 。

特殊性质 B：对于所有 $1 \leq i \leq n$, 均有 $0 \leq a_i \leq 1$ 。

特殊性质 C：对于所有 $1 \leq i \leq n$, 均有 $0 \leq a_i \leq 255$ 。

多边形 (polygon)

【题目描述】

小 R 喜欢玩小木棍。小 R 有 n 根小木棍，第 i ($1 \leq i \leq n$) 根小木棍的长度为 a_i 。

小 X 希望小 R 从这 n 根小木棍中选出若干根小木棍，将它们按任意顺序首尾相连拼成一个多边形。小 R 并不知道小木棍能拼成多边形的条件，于是小 X 直接将条件告诉了他：对于长度分别为 l_1, l_2, \dots, l_m 的 m 根小木棍，这 m 根小木棍能拼成一个多边形当且仅当 $m \geq 3$ 且所有小木棍的长度之和大于所有小木棍的长度最大值的两倍，即 $\sum_{i=1}^m l_i > 2 \times \max_{i=1}^m l_i$ 。

由于小 R 知道了小木棍能拼成多边形的条件，小 X 提出了一个更难的问题：有多少种选择小木棍的方案，使得选出的小木棍能够拼成一个多边形？你需要帮助小 R 求出选出的小木棍能够拼成一个多边形的方案数。两种方案不同当且仅当选择的小木棍的下标集合不同，即存在 $1 \leq i \leq n$ ，使得其中一种方案选择了第 i 根小木棍，但另一种方案未选择。由于答案可能较大，你只需要求出答案对 998,244,353 取模后的结果。

【输入格式】

从文件 *polygon.in* 中读入数据。

输入的第一行包含一个正整数 n ，表示小 R 的小木棍的数量。

输入的第二行包含 n 个正整数 a_1, a_2, \dots, a_n ，表示小 R 的小木棍的长度。

【输出格式】

输出到文件 *polygon.out* 中。

输出一行一个非负整数，表示小 R 选出的小木棍能够拼成一个多边形的方案数对 998,244,353 取模后的结果。

【样例 1 输入】

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

【样例 1 输出】

```
1 9
```

【样例 1 解释】

共有以下 9 种选择小木棍的方案，使得选出的小木棍能够拼成一个多边形：

1. 选择第 2, 3, 4 根小木棍，长度之和为 $2 + 3 + 4 = 9$ ，长度最大值为 4；
2. 选择第 2, 4, 5 根小木棍，长度之和为 $2 + 4 + 5 = 11$ ，长度最大值为 5；
3. 选择第 3, 4, 5 根小木棍，长度之和为 $3 + 4 + 5 = 12$ ，长度最大值为 5；
4. 选择第 1, 2, 3, 4 根小木棍，长度之和为 $1 + 2 + 3 + 4 = 10$ ，长度最大值为 4；
5. 选择第 1, 2, 3, 5 根小木棍，长度之和为 $1 + 2 + 3 + 5 = 11$ ，长度最大值为 5；
6. 选择第 1, 2, 4, 5 根小木棍，长度之和为 $1 + 2 + 4 + 5 = 12$ ，长度最大值为 5；
7. 选择第 1, 3, 4, 5 根小木棍，长度之和为 $1 + 3 + 4 + 5 = 13$ ，长度最大值为 5；
8. 选择第 2, 3, 4, 5 根小木棍，长度之和为 $2 + 3 + 4 + 5 = 14$ ，长度最大值为 5；
9. 选择第 1, 2, 3, 4, 5 根小木棍，长度之和为 $1 + 2 + 3 + 4 + 5 = 15$ ，长度最大值为 5。

【样例 2 输入】

```
1 5
2 2 2 3 8 10
```

【样例 2 输出】

```
1 6
```

共有以下 6 种选择小木棍的方案，使得选出的小木棍能够拼成一个多边形：

1. 选择第 1, 2, 3 根小木棍，长度之和为 $2 + 2 + 3 = 7$ ，长度最大值为 3；
2. 选择第 3, 4, 5 根小木棍，长度之和为 $3 + 8 + 10 = 21$ ，长度最大值为 10；
3. 选择第 1, 2, 4, 5 根小木棍，长度之和为 $2 + 2 + 8 + 10 = 22$ ，长度最大值为 10；
4. 选择第 1, 3, 4, 5 根小木棍，长度之和为 $2 + 3 + 8 + 10 = 23$ ，长度最大值为 10；
5. 选择第 2, 3, 4, 5 根小木棍，长度之和为 $2 + 3 + 8 + 10 = 23$ ，长度最大值为 10；
6. 选择第 1, 2, 3, 4, 5 根小木棍，长度之和为 $2 + 2 + 3 + 8 + 10 = 25$ ，长度最大值为 10。

【样例 3】

见选手目录下的 *polygon/polygon3.in* 与 *polygon/polygon3.ans*。

该样例满足测试点 7 ~ 10 的约束条件。

【样例 4】

见选手目录下的 *polygon/polygon4.in* 与 *polygon/polygon4.ans*。

该样例满足测试点 11 ~ 14 的约束条件。

【子任务】

对于所有测试数据，保证：

- $3 \leq n \leq 5,000$;
- 对于所有 $1 \leq i \leq n$ ，均有 $1 \leq a_i \leq 5,000$ 。

测试点编号	$n \leq$	$\max_{i=1}^n a_i \leq$
1 ~ 3	3	10
4 ~ 6	10	10^2
7 ~ 10	20	
11 ~ 14	500	
15 ~ 17		
18 ~ 20	5,000	1
21 ~ 25		5,000