2026 CCF 非专业级软件能力认证

CSP-J/S 2026 第二轮

入门组

时间：2026 年 10 月 31 日 08:30 ～ 12:00

题目名称	旅游	回文串	按位与	纸牌游戏
题目类型	传统型	传统型	传统型	传统型
目录	`travel`	`string`	`and`	`card`
可执行文件名	`travel`	`string`	`and`	`card`
输入文件名	`travel.in`	`string.in`	`and.in`	`card.in`
输出文件名	`travel.out`	`string.out`	`and.out`	`card.out`
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	1024 MiB	1024 MiB	1024 MiB	1024 MiB
测试点数目	20	25	20	25
测试点是否等分	是	是	是	是

提交程序源文件名

对于 C++ 语言	`travel.cpp`	`string.cpp`	`and.cpp`	`card.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. 程序可使用的栈空间内存限制与题目的内存限制一致。
6. 禁止在提交的程序源文件中改变编译器参数（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。
7. 提交的程序源文件的放置位置请参考各省的具体要求。
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。

旅游（travel）

【题目描述】

小 R 和小 X 准备坐高铁出门旅游。他们买了两张二等座车票，车票上写着各自的车厢号和座位号。两人希望座位相邻，这样旅途中可以坐在一起。

他们乘坐的列车由 17 节车厢组成，车厢编号是 `01` 到 `17`。其中，`01` 和 `16` 号车厢是一等座车厢，其余车厢是二等座车厢。两人买的都是二等座车票，座位都在二等座车厢中。每节二等座车厢共有 18 排，每排有 5 个座位。座位号由两位排号和一位大写字母组成，例如 `03C` 表示第 3 排 C 座。每一排的座位按照从左到右的顺序排列，座位字母依次是 `A`、`B`、`C`、`D`、`F`，其中 C 座与 D 座之间是过道，即每排过道左侧的座位序列是 `A`、`B`、`C`，右侧是 `D`、`F`。以下是一节二等座车厢的座位示意图：



两个座位相邻，当且仅当在同一节车厢的同一排，位于过道的同一侧，且在该侧的座位序列中相邻。

你需要帮助小 R 依次判断：

- 两张车票对应的座位是否在同一节车厢；
- 两张车票对应的座位是否在同一节车厢的同一排；
- 两张车票对应的座位是否在同一节车厢的同一排，且位于过道的同一侧；
- 两张车票对应的座位是否相邻。

【输入格式】

从文件 ***travel.in*** 中读入数据。

输入的第一行包含四个字符串 c_1, s_1, c_2, s_2，分别表示小 R 的车票上的车厢号、座位号，以及小 X 的车票上的车厢号、座位号。

【输出格式】

输出到文件 ***travel.out*** 中。

输出一行一个长度为 4 的 01 字符串，分别表示四个问题的答案，其中 1 代表是，0 代表否。

【样例 1 输入】

```
02 01A 02 01B
```

【样例 1 输出】

```
1111
```

【样例 1 解释】

02 号车厢是二等座车厢。01A 和 01B 在同一节车厢、同一排、过道的同一侧，并且相邻，因此输出 1111。

【样例 2 输入】

```
02 01C 02 01A
```

【样例 2 输出】

```
1110
```

【样例 2 解释】

两个座位在同一节车厢、同一排、过道的同一侧，但 01C 与 01A 中间隔着 01B，不相邻，因此输出 1110。

【样例 3 输入】

```
02 01C 02 01D
```

【样例 3 输出】

```
1100
```

【样例 4 输入】

```
02 02B 02 01A
```

【样例 4 输出】

```
1000
```

【样例 5 输入】

```
02 01A 03 01A
```

【样例 5 输出】

```
0000
```

【数据范围】

对于所有测试数据，均有：

- c_1, c_2 是 01 到 17 之间除 01 和 16 外的两位车厢号；
- s_1, s_2 是对应车厢中的合法座位号，即前两位是 01 到 18 之间的排号，末位是 A、B、C、D、F 中的一个字母；
- 两张车票对应的不是同一个座位，即 $c_1 \neq c_2$ 或 $s_1 \neq s_2$。

测试点编号	特殊性质
$1 \sim 4$	A, B, C
$5 \sim 10$	A, B
$11 \sim 14$	A
$15 \sim 17$	D
$18 \sim 20$	无

特殊性质 A：两个座位在同一节车厢，即 $c_1 = c_2$。

特殊性质 B：两个座位在同一排，即 s_1 的前两位与 s_2 的前两位相同。

特殊性质 C：两个座位在过道的同一侧。

特殊性质 D：小 R 的座位是 D 座，小 X 的座位是 F 座，即 s_1 的末位字符是 D，s_2 的末位字符是 F。

回文串（string）

【题目描述】

小 R 有一个长度为 L 的字符串 $S = s_1 \dots s_L$，其中只包含小写英文字母。小 R 发现，虽然 S 可能很长，但它可以分成 n 个连续的段，其中第 i $(1 \le i \le n)$ 段由 a_i 个小写英文字母 c_i 构成。把这 n 段依次拼接起来便得到 S，因此 $L = \sum_{i=1}^{n} a_i$。相邻两段中的字符可以相同。

小 X 喜欢回文串。若一个字符串反转后与原字符串相同，就称它是回文串，例如 `aba`、`abba` 都是回文串，而 `aabbc` 不是。也就是说，字符串 S 是回文串，当且仅当对于所有 i $(1 \le i \le L)$，均有 $s_i = s_{L+1-i}$。

小 X 想让小 R 把 S 改成回文串。小 R 可以将 S 中任意一个位置的字符修改成任意小写英文字母。他想用尽量少的修改满足小 X 的要求。

你需要帮助小 R 求出，他最少需要修改多少个位置的字符，才能将 S 改成回文串。

【输入格式】

从文件 ***string.in*** 中读入数据。

输入的第一行包含一个正整数 n。

输入的第 $i+1$ $(1 \le i \le n)$ 行包含一个正整数 a_i 和一个小写英文字母 c_i。

【输出格式】

输出到文件 ***string.out*** 中。

输出一行一个整数，表示小 R 最少需要修改多少个位置的字符，才能将 S 改成回文串。

【样例 1 输入】

```
3
2 a
2 b
1 c
```

【样例 1 输出】

```
2
```

【样例 1 解释】

小 R 的字符串 S 由 3 段构成：第 1 段由 2 个 a 组成，第 2 段由 2 个 b 组成，第 3 段由 1 个 c 组成，依次拼接得到 $S =$ aabbc。

小 R 可以将最后一个字符 c 和倒数第二个字符 b 都改为 a，得到回文串 aabaa，共修改 2 个位置的字符。可以证明，小 R 最少需要修改 2 个位置的字符，才能将 S 改成回文串。

【样例 2 输入】

```
2
2 a
3 a
```

【样例 2 输出】

```
0
```

【样例 2 解释】

小 R 的字符串 S 由 2 段构成：第 1 段由 2 个 a 组成，第 2 段由 3 个 a 组成，依次拼接得到 $S =$ aaaaa，则 S 已经是回文串，不需要修改。

【样例 3 输入】

```
6
1000000000 a
1000000000 a
1000000000 a
1000000000 b
1000000000 b
1000000000 b
```

【样例 3 输出】

```
3000000000
```

【样例 4】

见题目目录下的 ***string/4.in*** 与 ***string/4.ans***。
该样例满足测试点 $10 \sim 12$ 的约束条件。

【样例 5】

见题目目录下的 ***string/5.in*** 与 ***string/5.ans***。
该样例满足测试点 $13 \sim 16$ 的约束条件。

【样例 6】

见题目目录下的 ***string/6.in*** 与 ***string/6.ans***。
该样例满足测试点 $19 \sim 21$ 的约束条件。

【数据范围】

对于所有测试数据，均有：

- $1 \le n \le 2 \times 10^5$；
- 对于所有 i $(1 \le i \le n)$，均有 $1 \le a_i \le 10^9$，且 c_i 为小写英文字母。

测试点编号	$n \le$	$\max_{i=1}^{n} a_i \le$
1	1	5
$2 \sim 4$	2	
$5, 6$	10	1
$7, 8$		50
9	2 000	1
$10 \sim 12$		50
$13 \sim 16$		10^9
$17, 18$	2×10^5	1
$19 \sim 21$		500
$22 \sim 25$		10^9

按位与（and）

【题目描述】

小 R 有 n 个非负整数 $a_1, a_2, \ldots, a_n$。

小 X 给了小 R 一个正整数 m，并要求小 R 从这 n 个非负整数中选出**恰好** m 个，使得选出的 m 个数的**二进制按位与**的值最大。

你需要帮助小 R 求出，他能选出的 m 个数的按位与的最大值。

【输入格式】

从文件 ***and.in*** 中读入数据。

输入的第一行包含两个正整数 n, m。

输入的第二行包含 n 个非负整数 $a_1, a_2, \ldots, a_n$，依次表示小 R 的 n 个非负整数。

【输出格式】

输出到文件 ***and.out*** 中。

输出一行一个非负整数，表示小 R 能选出的 m 个数的按位与的最大值。

【样例 1 输入】

```
4 2
3 15 12 12
```

【样例 1 输出】

```
12
```

【样例 1 解释】

小 R 有 4 个非负整数：$3, 15, 12, 12$。

- 若小 R 选择 $3, 15$，则按位与的值为 $3 \text{ and } 15 = 3$；
- 若小 R 选择 $3, 12$，则按位与的值为 $3 \text{ and } 12 = 0$；
- 若小 R 选择 $15, 12$，则按位与的值为 $15 \text{ and } 12 = 12$；
- 若小 R 选择 $12, 12$，则按位与的值为 $12 \text{ and } 12 = 12$。

因此，小 R 能选出的 m 个数的按位与的最大值为 12。

【样例 2 输入】

```
5 3
31 3 1 15 7
```

【样例 2 输出】

```
7
```

【样例 2 解释】

若小 R 选择 $31, 15, 7$，则按位与的值为 $31 \operatorname{and} 15 \operatorname{and} 7 = 7$。可以证明，小 R 能选出的 m 个数的按位与的最大值为 7。

【样例 3】

见题目目录下的 ***and/3.in*** 与 ***and/3.ans***。
该样例满足测试点 $8 \sim 10$ 的约束条件。

【样例 4】

见题目目录下的 ***and/4.in*** 与 ***and/4.ans***。
该样例满足测试点 $14, 15$ 的约束条件。

【样例 5】

见题目目录下的 ***and/5.in*** 与 ***and/5.ans***。
该样例满足测试点 $18, 19$ 的约束条件。

【数据范围】

对于所有测试数据，均有：

- $1 \leq m \leq n \leq 2 \times 10^5$；
- 对于所有 i $(1 \leq i \leq n)$，均有 $0 \leq a_i < 2^{30}$。

测试点编号	$n \leq$	m	$\max_{i=1}^{n} a_i <$	特殊性质
$1 \sim 5$	20	$\leq n$	2^{20}	无
6	2 000	$= 1$		
7		$= n$		
$8 \sim 10$		$= 2$		
$11, 12$	2×10^5	$= n - 1$		
13		$\leq n$		A
$14, 15$				B
$16, 17$			4	无
$18, 19$			2^{10}	
20			2^{30}	

特殊性质 A：对于所有 i $(1 \leq i \leq n)$，均存在非负整数 k_i 满足 $a_i = 2^{k_i} - 1$。

特殊性质 B：存在非负整数 k 满足 $n = 2^k$，且对所有 i $(1 \leq i \leq n)$，均有 $a_i = i - 1$。

纸牌游戏（card）

【题目描述】

小 Z 正在玩一种单人纸牌游戏。

他有一叠特制的纸牌，共有 n 张，从上到下的点数依次为 $a_1, a_2, \ldots, a_n$。每张牌的点数都是 1 到 m 之间的整数，且每种点数都出现至少一次。

小 Z 将进行 n 局相互独立的游戏。每局开始前，他会将所有牌按原顺序放回牌堆，并清空桌面。在第 i $(1 \le i \le n)$ 局中，他跳过前 $i-1$ 张牌，依次打出第 i 张到第 n 张牌，这些牌的点数依次为 $a_i, a_{i+1}, \ldots, a_n$。

桌面上的牌始终从左到右排成一行。打出一张点数为 k 的牌时：

- 若桌面上没有点数为 k 的牌，就将这张牌放在最右端；
- 否则，收走桌面上点数为 k 的牌及其右侧的所有牌，再把这张牌放在最右端。被收走的牌在本局中不再使用。

由上述规则可知，任意时刻桌面上同一种点数的牌至多只有一张。

最后一张牌打出后，本局游戏结束。

例如，若桌面上从左到右的牌的点数依次为 $1, 4, 2$，则打出一张点数为 3 的牌后，桌面上从左到右的牌的点数依次为 $1, 4, 2, 3$。接着打出一张点数为 4 的牌时，桌面上原来点数为 4 的牌及其右侧的所有牌（点数依次为 $4, 2, 3$）被收走，再把新打出的 4 放在最右端，此时桌面上从左到右的牌的点数依次为 $1, 4$。

你需要帮助小 Z 依次求出，对于每个 $i = 1, 2, \ldots, m$，在所有 n 局游戏中，有多少局结束时桌面上留有点数为 i 的牌。

【输入格式】

从文件 ***card.in*** 中读入数据。

输入的第一行包含两个正整数 n, m，分别表示牌的数量和点数种类数。

输入的第二行包含 n 个正整数 $a_1, a_2, \ldots, a_n$，依次表示小 Z 的牌从上到下的点数。

【输出格式】

输出到文件 ***card.out*** 中。

输出一行 m 个非负整数，其中第 i $(1 \le i \le m)$ 个整数表示在所有 n 局游戏中，有多少局结束时桌面上留有点数为 i 的牌。

【样例 1 输入】

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

【样例 1 输出】

```
3 1 4
```

【样例 1 解释】

对于第 1 局游戏，小 Z 依次打出点数为 $1,2,1,3$ 的牌：

1. 打出第 1 张牌（点数为 1）后，桌面上只有一张点数为 1 的牌；
2. 打出第 2 张牌（点数为 2）后，桌面上从左到右的牌的点数依次为 $1,2$；
3. 打出第 3 张牌（点数为 1）后，先收走桌面上点数为 $1,2$ 的两张牌，再放上新打出的 1，桌面上只有一张点数为 1 的牌；
4. 打出第 4 张牌（点数为 3）后，桌面上从左到右的牌的点数依次为 $1,3$。

因此第 1 局游戏结束时，桌面上从左到右的牌的点数依次为 $1,3$。

对于第 2 局游戏，小 Z 依次打出点数为 $2,1,3$ 的牌，结束时桌面上从左到右的牌的点数依次为 $2,1,3$。

对于第 3 局游戏，小 Z 依次打出点数为 $1,3$ 的牌，结束时桌面上从左到右的牌的点数依次为 $1,3$。

对于第 4 局游戏，小 Z 仅打出一张点数为 3 的牌，结束时桌面上只有一张点数为 3 的牌。

因此，在全部 4 局游戏中：

- 有 3 局游戏结束时，桌面上留有点数为 1 的牌，即第 $1,2,3$ 局；
- 有 1 局游戏结束时，桌面上留有点数为 2 的牌，即第 2 局；
- 有 4 局游戏结束时，桌面上留有点数为 3 的牌，即全部 4 局。

【样例 2 输入】

```
10 3
3 1 2 1 3 2 3 1 2 1
```

【样例 2 输出】

```
10 3 3
```

【样例 3】

见题目目录下的 ***card/3.in*** 与 ***card/3.ans***。

该样例满足测试点 $4 \sim 6$ 的约束条件。

【样例 4】

见题目目录下的 ***card/4.in*** 与 ***card/4.ans***。
该样例满足测试点 $7 \sim 9$ 的约束条件。

【样例 5】

见题目目录下的 ***card/5.in*** 与 ***card/5.ans***。
该样例满足测试点 $15 \sim 17$ 的约束条件。

【样例 6】

见题目目录下的 ***card/6.in*** 与 ***card/6.ans***。
该样例满足测试点 $21, 22$ 的约束条件。

【数据范围】

对于所有测试数据，均有：

- $2 \le n \le 5 \times 10^5$，$1 \le m \le 3 \times 10^5$；
- 对于所有 i $(1 \le i \le n)$，均有 $1 \le a_i \le m$；
- 每种点数都出现至少一次，即对于所有 i $(1 \le i \le m)$，均存在 j $(1 \le j \le n)$ 满足 $a_j = i$。

测试点编号	$n \le$	$m \le$	特殊性质
$1 \sim 3$	5	2	无
$4 \sim 6$	500	200	
$7 \sim 9$	5 000	3 000	A
$10 \sim 12$			无
$13, 14$	2×10^5	2	
$15 \sim 17$		3 000	B
$18, 19$			无
20		10^5	A
$21, 22$			无
$23 \sim 25$	5×10^5	3×10^5	

特殊性质 A：$a_1 \le a_2 \le \cdots \le a_n$。
特殊性质 B：至多存在 4 999 个 i $(1 \le i \le n-1)$ 满足 $a_i \ne a_{i+1}$。