

2025 CCF 非专业软件能力认证

CSP-J/S 2025 第二轮认证

时间：2025 年 10 月 3 日 14:00 ~ 17:00

题目名称	山丘	字符对字符错	旅行	冰淇淋
题目类型	传统题	传统题	传统题	传统题
目录	mountain	string	travel	ice
可执行文件名	mountain	string	travel	ice
输入文件名	mountain.in	string.in	travel.in	ice.in
输出文件名	mountain.out	string.out	travel.out	ice.out
测试点时限	1.0 秒	1.0 秒	3.0 秒	2.0 秒
内存限制	512 MB	512 MB	512 MB	512MB
子任务数目	10	10	5	5
是否捆绑测试	否	否	是	是

提交源程序文件名

对于 C++	mountain.cpp	string.cpp	travel.cpp	ice.cpp
--------	--------------	------------	------------	---------

编译选项

对于 C++	-lm -std=c++14 -O2 -Wl,--stack=998244353
--------	--

注意事项与提醒（请选手务必仔细阅读）

1. 选手提交的源程序必须存放在已建立好的，且带有**样例文件**和**下发文件**的的文件夹中，文件名称与对应试题英文名一致；
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 `0`。
4. **对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。**
5. 若无特殊说明，结果比较方式为**忽略行末空格、文末回车后的全文比较**。。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 若无特殊说明，每道题的**代码大小限制为 100KB**。
9. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔。
10. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。

11. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。
12. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。
13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。

山丘 (mountain)

【题目描述】

翁老师远行时看到了山丘，然后把它画了下来。
画由三种字符 `.` `/` `\` 组成。一座山的样子可能如下所示：

```
1 .....  
2 .. /\ .. .. /\ .. .. /\ ..  
3 ./... \ . ./... \ . ./... \ .  
4 /.... \ ..... /....
```

- 如果某个位置是 `/`，它的左下角可能是 `.` 或 `/`
- 如果某个位置是 `\`，它的右下角可能是 `.` 或 `\`
- `.` 表示空气，`/` 和 `\` 表示山体，一行字符中有且只有一个 `/` 或 `\`
- `/\` 表示山顶，**每座山有且仅有一个山顶**（没有山顶的不判定为山）。

现在翁老师想考考你，这里一共有几座山？

【输入格式】

第一行输入一个正整数 T ，表示测试数据组数。
对于每组测试数据：

- 第一行两个正整数 n, m ，表示字符画的大小是 $n \times m$ 的。
- 接下来 n 行，每行 m 个连续的字符，由三种 `.` `/` `\` 组成。

【输出格式】

对于每组测试数据：输出一行一个整数 k ，表示答案。

【样例 1 输入】

```
1 2
2 5 10
3 .....
4 .....
5 .../\.....
6 ../\.....
7 \/\.....\//
8 5 7
9 .....
10 .....
11 /\.....\//
12 ..\/\.....
13 .....
```

【样例 1 输出】

```
1 2
2 2
```

【样例 2】

见选手目录下的 mountain/mountain2.in 与 mountain/mountain2.ans

【数据范围与提示】

对于 100% 的数据，保证 $1 \leq T \leq 5 \times 10^3$ ， $1 \leq \sum n \times m \leq 5 \times 10^5$ 。

字符对字符错 (string)

【题目描述】

在一个只有大写字母的字符串 s 中，如果存在位置 $i < j$ 且 $s[i] = s[j]$ ，则称 (i, j) 为字符串 s 中的一个 **字符对**。

然而，如果存在位置 $i < j$ 且 $s[i] \neq s[j]$ ，则称 (i, j) 为字符串 s 中的一个 **字符错**。

翁老师的字符串有点不一样，它可能含有一些字符 $?$ 。

- 当字符 $?$ 出现时，它有可能是 **任何一个** 大写字母（同一时刻只能看作一个）。

现在给你一个字符串，请你求出这个字符串中不同的 **字符错** 的 **最大个数**。

对于两个 **字符错**，只要它们的 i 位置不同 **或** j 位置不同，我们就认为这两个 **字符错** 是 **不同的**。

此外，翁老师为了检验你对此题的理解程度，他会询问你 t 次。

【输入格式】

第一行一个整数 t ，表示询问次数。

- 接下来 t 行每行一个字符串 s 。 s 只包含大写英文字母和 $?$ 。

【输出格式】

输出 t 行，每行一个整数，表示不同字符错的最大个数。

【样例 1 输入】

```
1 1
2 ???CCF
```

【样例 1 输出】

```
1 20
```

【样例 1 解释】

在这个长度为 7 的字符串中，一共有 21 对字符。并且已经存在一对字符对即 (C, C) 。

可以证明通过合理的给 $?$ 设定适当的字符，使得字符错的最大个数为 $21 - 1 = 20$ 。

【样例 2 输入】

```
1 3
2 YIYISIWUYISI
3 YIJIUYIJIUBAYILING
4 YIYISIWUYISIIYIJIUYIJIUBAYILING
```

【样例 2 输出】

```
1 52
2 133
3 360
```

【样例 3 输入】

```
1 5
2 NI?IQ?O?CRXO?XQH?J?R
3 ??ZPJ??O?Z??ZPPVQ?I?
4 INUX??AOJKI?YJYWVF?I
5 I?FM?X?O?QC???XUX?KI
6 ?F??L??GTB??QWYBIUOQ
```

【样例 3 输出】

```
1 185
2 184
3 185
4 186
5 188
```

【样例 4】

见选手目录下的 string/string4.in 与 string/string4.ans

【样例 5】

见选手目录下的 string/string5.in 与 string/string5.ans

【样例 6】

见选手目录下的 string/string6.in 与 string/string6.ans

【数据范围与提示】

设 $|s|$ 表示字符串的长度。

- 对于 20% 的数据, $|s| \leq 100$ 。
- 对于 40% 的数据, $|s| \leq 1000$
- 对于另外 20% 的数据, 所有的 s 中都不包含 ?。
- 对于另外 10% 的数据, s 只包含 ?。
- 对于 100% 的数据, $1 \leq t \leq 10, 1 \leq |S| \leq 10^6$

旅行 (travel)

【题目描述】

西西艾弗市为了方便游客的交通出行，销售以下几种交通票券。

类别	使用人数	有效期	价格	备注
1 日票	1 人	购买当天，共 1 天	p_1	有效期内仅限购买者本人使用
3 日票	1 人	含购买当天在内的连续 3 天	p_3	有效期内仅限购买者本人使用
5 日票	1 人	含购买当天在内的连续 5 天	p_5	有效期内仅限购买者本人使用
组合票	2 人	含购买当天在内的连续 4 天	p_{pair}	有效期内两人均可使用

所有票券均在 **购买后立即生效**，并可在票券上标明的有效期内使用交通工具。（例如组合票购买后两个人就可以同时使用了，而不能一个人先用另一个人攒着后面再用）

当然，即使持有票券但未使用交通工具，或持有多张有效期重叠的票券，或票券的有效期超出了 n 天的观光行程也都是允许的。另外请注意， $p_1 \leq p_3 \leq p_5$ 这一关系 **并非总是成立**。

翁老师和聪聪老师将在西西艾弗市一同停留 n 天。但是，两人各自制定了自己的观光计划，并决定了每天自己是否要进行观光。为了完成观光行程，对于每个人，在他们**进行观光的每一天**，都必须持有一张有效的票券（包括组合票）。

翁老师的日程由字符串 $a = a_1a_2 \cdots a_n$ 表示，聪聪老师的日程由字符串 $b = b_1b_2 \cdots b_n$ 表示时，对于日期 $i(1 \leq i \leq n)$:

- 如果翁老师进行观光， $a_i = 1$ ；否则 $a_i = 0$
- 如果聪聪老师进行观光， $b_i = 1$ ；否则 $b_i = 0$

请根据以上形式给出的日程，编写一个程序，计算出为了确保在每个人进行观光的每一天都持有至少一张有效票券（包括组合票）所需的最少费用。

【输入格式】

第一行给定一个表示在西西艾弗市停留时间的整数 n 。
第二行给定一个表示翁老师日程的字符串 a 。
第三行给定一个表示聪聪老师日程的字符串 b 。
第四行给定 p_1, p_3, p_5, p_{pair} ，以空格分隔。

【输出格式】

在第一行输出表示最小费用的整数。

【样例输入】

```
1 9
2 011011101
3 110001110
4 3 7 12 15
```

【样例输出】

```
1 29
```

【样例 1 解释】

$n = 9, p_1 = 3, p_3 = 7, p_5 = 12, p_{pair} = 15$ ，翁老师和聪聪老师各自的日程如下：

日期	1	2	3	4	5	6	7	8	9
翁老师	0	1	1	0	1	1	1	0	1
聪聪老师	1	1	0	0	0	1	1	1	0

如果只使用 1 日票，总观光天数为 11 天 (翁老师 6 天 + 聪聪老师 5 天)，费用为 11 (观光天数) $\times$ 3 (1 日票价格) = 33。

但是，如果两人在 **第 5 天至第 8 天** 共享一张组合票，总费用仅为 30。

也可以翁老师购买一张 **第 5 天至第 7 天** 的 3 日票，聪聪老师购买一张 **第 6 天至第 8 天** 的 3 日票，总费用可以节省至 29。

【样例 2 输入】

```
1 9
2 011011101
3 110001110
4 1 10000 10000 10000
```

【样例 2 输出】

```
1 11
```

【样例 3】

见选手目录下的 travel/travel3.in 与 travel/travel3.ans

【样例 4】

见选手目录下的 travel/travel4.in 与 travel/travel4.ans

【样例 5】

见选手目录下的 travel/travel5.in 与 travel/travel5.ans

【样例 6】

见选手目录下的 travel/travel6.in 与 travel/travel6.ans

【样例 7】

见选手目录下的 travel/travel7.in 与 travel/travel7.ans

【数据范围与提示】

对于 100% 的数据满足： $1 \leq n \leq 2000$ ， $1 \leq p_1, p_3, p_5, p_{pair} \leq 10000$ 。

本题采用捆绑测试

子任务编号	特殊限制	分值
1	$p_1 = 1, p_3 = p_5 = p_{pair} = 10000$	6
2	$p_{pair} = 1, p_1 = p_3 = p_5 = 10000$	12
3	所有 $a_i = b_i = 1$	16
4	所有 $b_i = 0$	24
5	无	42

冰淇淋 (ice)

【题目描述】

翁老师和聪聪老师来到了冰淇淋店。这家店的顾客可以通过选择一种口味、一种蛋筒和一种配料来订购冰淇淋。

- 口味有 x 种，价格分别为 $a_1, a_2, \dots, a_x$ 。
- 蛋筒有 y 种，价格分别为 $b_1, b_2, \dots, b_y$ 。
- 配料有 z 种，价格分别为 $c_1, c_2, \dots, c_z$ 。

冰淇淋的价格是所选口味、蛋筒和配料价格的总和。给定一个整数 p ，冰淇淋的得分定义为价格与 p 之差的绝对值。

- 例如：假设选择的口味，蛋筒，配料的编号分别为 i, j, k 。则得分为 $|a_i + b_j + c_k - p|$ 。

翁老师和聪聪老师想要一起订购一个冰淇淋，但他们对冰淇淋的选择目标完全相反。具体来说，翁老师希望最大化得分，而聪聪老师希望最小化得分。因此，他们决定按照以下方式选择冰淇淋的口味、蛋筒和配料：

1. 首先，翁老师选择口味。
2. 然后，聪聪老师选择蛋筒。
3. 最后，翁老师选择配料。

给定口味、蛋筒、配料的信息以及整数 p ，编写一个程序，计算当双方都采取最佳策略时，最终订购的冰淇淋的得分。

【输入格式】

第一行输入四个空格隔开的整数 x, y, z, p 。

接下来一行输入 $a_1, a_2, \dots, a_x$ 。

接下来一行输入 $b_1, b_2, \dots, b_y$ 。

接下来一行输入 $c_1, c_2, \dots, c_z$ 。

【输出格式】

输出最终订购的冰淇淋的得分。

【样例 1 输入】

1	1 1 3 22
2	5

3	10
4	9 2 3

【样例 1 输出】

1	5
---	---

【样例 1 解释】

- 口味价格为 5。
- 蛋筒价格为 10。
- 配料价格分别为 9、2、3。

翁老师首先选择价格为 5 的口味，聪聪老师选择价格为 10 的蛋筒。最后，翁老师选择价格为 2 的配料，使得总价格为 17，得分为 $|17 - 22| = 5$ 。

【样例 2 输入】

1	1 2 2 100
2	11
3	33 44
4	40 60

【样例 2 输出】

1	15
---	----

【样例 2 解释】

- 口味价格为 11。
- 蛋筒价格分别为 33、44。
- 配料价格分别为 40、60。

翁老师选择价格为 11 的口味，聪聪老师选择价格为 44 的蛋筒（因为这样可以使翁老师选择价格为 60 的配料，得分为 $|115 - 100| = 15$ ）。

注意：聪聪老师不会选择价格为 33 的蛋筒，因为这样不会使得得分最小化。

【样例 3 输入】

```
1 2 2 2 0
2 15 23
3 5 16
4 23 45
```

【样例 3 输出】

```
1 73
```

【样例 3 解释】

- 口味价格分别为 15、23。
- 蛋筒价格分别为 5、16。
- 配料价格分别为 23、45。

翁老师选择价格为 23 的口味，聪聪老师选择价格为 5 的蛋筒，翁老师选择价格为 45 的配料，总价格为 73，得分为 $|73 - 0| = 73$ 。

【样例 4 输入】

```
1 3 3 3 50
2 12 5 5
3 2 19 37
4 10 5 15
```

【样例 4 输出】

```
1 14
```

【样例 4 解释】

- 口味价格分别为 12、5、5。
- 蛋筒价格分别为 2、19、37。
- 配料价格分别为 10、5、15。

翁老师选择价格为 12 的口味，聪聪老师选择价格为 2 的蛋筒，翁老师选择价格为 15 的配料，总价格为 29，得分为 $|29 - 50| = 21$ 。然而，聪聪老师会选择价格为 19 的蛋筒，使得翁老师选择价格为 15 的配料，总价格为 46，得分为 $|46 - 50| = 4$ 。但最终得分为 14，因为翁老师会选择最优策略。

【样例 5】

见选手目录下的 ice/ice5.in 与 ice/ice5.ans

【样例 6】

见选手目录下的 ice/ice6.in 与 ice/ice6.ans

【样例 7】

见选手目录下的 ice/ice7.in 与 ice/ice7.ans

【样例 8】

见选手目录下的 ice/ice8.in 与 ice/ice8.ans

【样例 9】

见选手目录下的 ice/ice9.in 与 ice/ice9.ans

【数据范围与提示】

对于 100% 的数据满足： $1 \leq x, y, z \leq 2 \times 10^5$ ， $1 \leq a_i, b_j, c_k \leq 10^8$ ， $0 \leq p \leq 3 \times 10^8$ 。

- 其中 $i \in [1, x]$ ， $j \in [1, y]$ ， $k \in [1, z]$ 。

本题采用捆绑测试

子任务编号	特殊限制	分值
1	$x = 1, y = 1, z \leq 100$	7
2	$x = 1, y \leq 100, z \leq 100$	17
3	$x, y, z \leq 100$	21
4	$x, y, z \leq 4000$	22
5	无额外约束	33