

2024 西安市信息学算法编程大赛

小高组复赛

时间：2024 年 4 月 27 日 09:00 ~ 12:00

题目名称	交替字符串	倒豆子	劳动最光荣	吃饭
题目类型	传统型	传统型	传统型	传统型
目录	alternate	bean	work	eat
可执行文件名	alternate	bean	work	eat
输入文件名	alternate.in	bean.in	work.in	eat.in
输出文件名	alternate.out	bean.out	work.out	eat.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
测试点题目	10	10	10	10
测试点是否等分	是	是	是	是

提交源文件程序名

alternate.cpp	bean.cpp	work.cpp	eat.cpp
---------------	----------	----------	---------

编译选项

-O2 -std=c++14

注意事项（请仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- C/C++ 中函数 main() 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。
- 提交的程序代码文件的放置位置请参考考场要求。
- 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
- 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）。
- 选手提交的程序源文件必须不大于 100 KB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。

交替字符串 (alternate)

【题目描述】

给定一个只包含两种字符的字符串 s ，如果 s 的一个子串中不存在两个相邻的字符相同的情况，则认为这是一个交替子字符串。

需要注意的是，两个子串起始位置不同，终止位置不同，则认为他们是不同的子串。

求字符串 s 的交替子字符串的个数。

【输入格式】

从文件 `alternate.in` 中读入数据。

一行，一个字符串 s ，仅包含两种字符。

【输出格式】

输出到文件 `alternate.out` 中。

一个整数，表示字符串 s 的交替子字符串的个数。

【样例 1 输入】

abbb

【样例 1 输出】

5

【样例解释 1】

共有“a”，“ab”，“b”，“b”，“b”五种不同的交替子串

【数据范围】

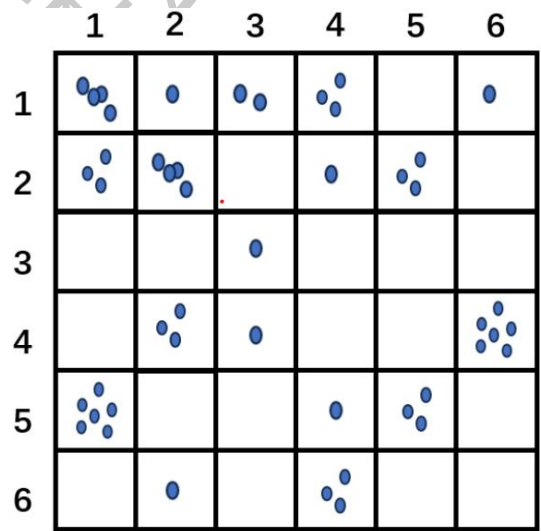
对于 30% 的数据， $1 \leq s.size() \leq 10^3$ 。

对于 100% 的数据， $1 \leq s.size() \leq 10^6$ 。

保证输入数据只出现大写字母和小写字母且字符串中有且仅有两种字符。

倒豆子 (bean)

【题目描述】



如图有一个正方形置物盘，置物盘上有 $n \times n$ 个小格子，每个小格子里有数量不等的豆子。对于整个置物盘，我们可以做以下两种操作之一：

操作一：选择置物盘中一行小格子，将这一行小格子中的豆子倒到相邻行中；再选择置物盘中一列小格子，将这一列小格子中的豆子倒到相邻列中。

操作二：将整个置物盘沿顺时针转动一圈。

假设需要做 m 次操作，请你选择最佳的操作过程，使得做完操作后，将其中的一个格子里的豆子拿出来作为种子时，种子的数量最多。

【输入格式】

从文件 `bean.in` 中读入数据。

总共 $n+2$ 行。

第一行一个正整数 n ，代表置物盘上的行数和列数。

第二到 $n+1$ 行每行 n 个非负整数 a_{ij} ，代表置物盘上第 i 行第 j 列的小格子里有 a_{ij} 个豆子。

第 $n+2$ 行一个正整数 m ，代表操作的次数。

【输出格式】

输出到文件 `bean.out` 中。

一个正整数，代表最终拿出来的豆子的数量。

【样例 1 输入】

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

【样例 1 输出】

```
10
```

【样例 2 输入】

```
6
4 1 2 3 0 1
3 4 0 1 3 0
0 0 1 0 0 0
0 3 1 0 0 6
6 0 0 1 3 0
0 1 0 3 0 0
1
```

【样例 2 输出】

```
12
```

【样例解释 1】

将第 1 行的豆子倒到第 2 行，再将第 1 列的豆子倒到第 2 列，此时第 2 行第 2 列的豆子数量是 10。

【样例解释 2】

（该样例如题目中的图片所示）将第 1 行的豆子倒到第 2 行，再将第 1 列的豆子倒到第 2 列，此时第 2 行第 2 列的豆子数量是 12。

【数据范围】

对于 20% 的数据， $2 \leq n \leq 10$, $1 \leq m \leq 10$, $0 \leq a_{i,j} \leq 100$ 。

对于 30% 的数据， $2 \leq n \leq 100$, $1 \leq m \leq 10$, $0 \leq a_{i,j} \leq 100$ 。

对于 100% 的数据， $2 \leq n \leq 10^3$, $1 \leq m \leq 10$, $0 \leq a_{i,j} \leq 10^8$ 。

特殊数据：对于 20% 的数据，保证 $m = 1$ 。

劳动最光荣 (work)

【题目描述】

小 L 所在的学校共有 n 名同学，每名同学在劳动节这一天被分配了基础劳动任务量，学校要求每名学生必须要达到 m 的劳动量才能获得“劳动之星”奖章。

但是很多同学以基础劳动量是不能够获得“劳动之星”的，所以学校允许自行组队以进行劳动量的分配， n 名学生的其中一部分学生将以团体的形式组队活动，每名学生至多只能加入 1 个团队。此时团队中个人的劳动量将变为他们团队总计劳动量的平均数。

例如，将初始的劳动量记为 $[4, 1, 3, 1]$ ，如果第 1 名同学和第 3 名同学自行组队，这两名同学的劳动总量为 $4 + 3 = 7$ ，然后将 $7 / 2 = 3.5$ 的劳动量平均分配给他们两人。因此，劳动量变为 $[3.5, 1, 3.5, 1]$ 。

由于学生众多，信息量巨大，所以学校不知道进行了多少次组队，以及组队的对象都是谁，请你计算出在若干次组队后，获得“劳动之星”奖章的同学最大的可能数量。

【输入格式】

从文件 `work.in` 中读入数据。

总共两行。

第 1 行包含两个整数 n, m 。分别表示学生的数量和获得“劳动之星”的劳动量。

第 2 行包含 n 个整数，表示全校学生的基础劳动量。

【输出格式】

输出到文件 `work.out` 中。

一个整数，表示最大可能能获得“劳动之星”学生的数量。

【样例 1 输入】

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

【样例 1 输出】

```
2
```

【样例 2 输入】

```
3 7
9 4 9
```

【样例 2 输出】

```
3
```

【样例解释 1】

按照题目中描述，重新分配劳动量为 $[3.5, 1, 3.5, 1]$ ，最多可能有 2 名同学获得“劳动之星”。

【样例解释 2】

所有的学生全体参与组队，重新分配劳动量后劳动量为 $[7\frac{1}{3}, 7\frac{1}{3}, 7\frac{1}{3}]$ ，最多可能有 3 名同学获得“劳动之星”。

【数据范围】

设第 i 名同学的劳动量为 a_i ：

对于 10% 的数据， $1 \leq n \leq 10^3, 1 \leq m \leq 10^4, 1 \leq a_i \leq 10^3$ 。

对于 100% 的数据， $1 \leq n \leq 10^5, 1 \leq m \leq 10^9, 1 \leq a_i \leq 10^9$ 。

吃饭 (eat)

【题目描述】

小明饿了, 所以小明要去吃饭。桌子上摆了 n 道菜, 第 i 道菜的编号是 i , 美味程度是 a_i 。小明可以选择从任意一道菜开始吃 (也可以不吃离场)。

由于餐厅的额外要求: 假设小明吃的上一道菜的编号是 x , 那么小明吃的下一道菜的编号必须是 x 的倍数。

问: 小明吃到的菜的美味程度之和最多是多少。

【输入格式】

从文件 `eat.in` 中读入数据。

总共两行。

第一行输入 n 。

第二行输入 n 个整数, 表示 a_1, \dots, a_n 。

【输出格式】

输出到文件 `eat.out` 中。

一个整数, 表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
7
```

【样例 2 输入】

```
5
1 -1 4 7 5
```

【样例 2 输出】

```
8
```

【样例解释 1】

小明先吃 1, 再吃 2, 再吃 4, 可以得到 $1+2+4=7$ 的美味程度。

【样例解释 2】

小明先吃 1, 再吃 4, 可以得到 $1+7=8$ 的美味程度。

【数据范围】

对于 20% 的数据, $n \leq 20$ 。

对于 30% 的数据, $n \leq 30$ 。

对于 60% 的数据, $n \leq 10000$ 。

对于 100% 的数据, $n \leq 2 \times 10^5$, $-10^5 \leq a_i \leq 10^5$ 。

特殊数据: 对于额外 20% 的数据, 保证 $a_i = i$ 。