2025 年 CSP-X 第二轮试题

题目名称	垃圾分类	昵称规范	国王模拟器	事件通报
题目类型	传统型	传统型	传统型	传统型
可执行文件名	rubbish.exe	name.exe	king.exe	report.exe
输入文件名	rubbish.in	name.in	king.in	report.in
输出文件名	rubbish.out	name.out	king.out	report.out
源程序文件名	rubbish.cpp	name.cpp	kign.cpp	report.cpp
每个测试点时限	1000ms	1000ms	1000ms	1000ms
内存限制	256MiB	256MiB	256MiB	256MiB
测试点数目	10	10	10	20
测试点是否等分	是	是	是	是

注意事项

1．代码必须放在子文件夹内，子文件夹名与题目英文名一致。文件名（包括程序名和输入输出文件名）必须使用英文小写。

2．C++ 编译选项：-O2 -std=c++14。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。

3．若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格分隔。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。

4．选手提交的程序源文件不能大于 100KB。

5．程序使用的栈空间内存限制与题目的内存限制要求一致。

垃圾分类（rubbish）

【题目描述】

2077 年，由于资源几近枯竭，梦之城推行了一套极其严格的垃圾分类制度。具体地，梦之城将垃圾分为 n 类，每一类垃圾只能被放入特定的垃圾桶中。由于梦之城掌握了压缩技术，因此在这里垃圾只有数量之分，没有体积大小之分。

你是梦之城的一位居民。在你居住的社区外有 n+1 个垃圾桶，标号为 1,2,...,n+1。对前 n 个垃圾桶，它们只能接受对应标号的垃圾，并且有一定的容量。第 i 个垃圾桶只能接受第 i 类垃圾，且最多只能被放入 r_i 个。对最后一个垃圾桶，它可以接受所有种类的垃圾，容量也是几近无限的。但是，每向这个垃圾桶放入一个垃圾，居委会会向你收取 c 的费用。

某一天，你的家中堆放满了垃圾。在将这些垃圾分类好后，你得到了一个长度为 n 的序列 $a_1,a_2,...,a_n$，其中 a_i 代表第 i 类垃圾的数量。你想要知道，如果想要扔掉所有的这些垃圾，你的最小花费是多少。

【输入格式】

从文件 rubbish.in 中读入数据。

第一行：两个整数 n,c，分别表示垃圾的种类数和向最后一个垃圾桶放入垃圾的费用。

第二行：n 个整数 $r_1,r_2,...,r_n$，分别表示前 n 个垃圾桶的最大容量。

第三行：n 个整数 $a_1,a_2,...,a_n$,，分别表示每一类垃圾的数量。

【输出格式】

输出到文件 rubbish.out 中。

输出一个整数，表示最小花费。

【样例 1 输入】

```
2 7
4 3
7 9
```

【样例 1 输出】

```
63
```

【样例 1 解释】

在最优情况下，需要向最后一个垃圾桶中放入 9 个垃圾，总花费 9×7=63 元。

【样例 2 输入】

```
2 10000
100 100
3 7
```

【样例 2 输出】

```
0
```

【样例 1 解释】

最优情况下，你不需要向最后一个垃圾桶中放入任何垃圾，费用为 0。

【数据范围】

对于前 20%的数据，$n=2$。

对于前 60%的数据，$n,a_i,c\leqslant 1000$。

对于另外 10%的数据，$c=0$。

对于另外 10%的数据，$r_i\geqslant a_i$。

对于 100%的数据，$2\leqslant n\leqslant 10^6$，$0\leqslant r_i,a_i,c\leqslant 10^6$。

昵称规范（name）

【题目描述】

火焰车最近在参加一个编程比赛，提交代码时需要给自己的答案取一个符合规则的昵称。他想出了一个字符串 S，由大小写英文字母组成，但比赛系统对昵称有严格要求：

- 昵称必须以大写字母 A 开头
- 从第 3 个字符到倒数第 2 个字符之间（包括端点），必须有且仅有一个大写字母 C。
- 除去开头的 A 和这个唯一的 C，其他所有字母必须是小写，显得低调又规范。

火焰车需要你帮他检查这个昵称 S 是否符合规则。如果符合，他就提交并标记为 AC（Accepted）；如果不符合，就标记为 WA（Wrong Answer）。你能帮他判断吗？

【输入格式】

从文件 name.in 中读入数据。

本题包含多组输入。

第一行：输入一个整数 T，表示输入数据组数。

第 2~n+1 行：每行输入一个字符串，仅由大小写英文字母组成。

【输出格式】

输出到文件 name.out 中。

对于输入的每行字符串，输出一个字符串答案，如果 s 满足所有条件，则输出 AC；否则输出 WA。

【样例 1 输入】

```
2
AcCcliw
AtCoCo
```

【样例 1 输出】

```
AC
WA
```

【样例 1 解释】

第一个数据符合所有条件；第二个数据 C 的数量超过了 1。

【样例 2 输入】

```
3
AabcdC
BaaCaa
AbCac
```

【样例 2 输出】

```
WA
WA
AC
```

【样例 2 解释】

第一个数据中，C 的位置不在合法范围内；第二个数据没有以 A 开头；第三个数据符合所有条件。

【数据范围】

对于 50%的数据，$4 \leqslant |s| \leqslant 10$，其中一半的数据保证 s 中有且仅有两个大写字母。

对于 100%的数据，$1 \leqslant T \leqslant 10$，$4 \leqslant |s| \leqslant 10^5$，s 的每个字符均为大写或小写字母。

其中|s|表示 s 的长度。

国王模拟器（king）

【题目描述】

在一款游戏中，你扮演一位国王。你的王国被划分为 n 个地区，第 i 个地区收藏着 a_i 件珍宝。你不喜欢住在宫殿里，每年你都会到下一个地区去视察，并在那里居住和生活。

被你视察过的地区会被你的霸王之气所折服。当某个地区被视察后，该地区会从下一年开始，每年将当地的一件珍宝供奉到你当前所在的地区（若无剩余则忽略）。

请你计算：n 年后，当你视察完全部的地区时，每个地区剩余的珍宝数量。

【输入格式】

从文件 king.in 中读入数据。

第一行：输入一个整数 n，表示地区数量。

第二行：输入 n 个整数 $a_1,a_2,...,a_n$，分别表示每个地区初始时的珍宝数量。

【输出格式】

输出到文件 king.out 中。

输出 n 个整数，分别表示每个地区最终的珍宝数量，以空格分隔。

【样例 1 输入】

```
4
1 2 3 4
```

【样例 1 输出】

```
0 1 3 6
```

【样例 1 解释】

第一年：在 1 号地区视察，每个地区的珍宝数量分别为 1,2,3,4。

第二年：在 2 号地区视察，1 号地区会供奉一件珍宝到 2 号地区。此时每个地区的珍宝数量分别为 0,3,3,4。

第三年：在 3 号地区视察，2 号地区会供奉一件珍宝到 3 号地区。此时每个地区的珍宝数量分别为 0,2,4,4。

第四年：在 4 号地区视察，2,3 号地区会各自供奉一件珍宝到 4 号地区。此时每个地区的珍宝数量分别为 0,1,3,6。

【样例 2 输入】

```
3
1 0 0
```

【样例 2 输出】

```
0 0 1
```

【样例 2 解释】

第一年：在 1 号地区视察，每个地区的珍宝数量分别为 1,0,0。

第二年：在 2 号地区视察，1 号地区会供奉一件珍宝到 2 号地区。此时每个地区的珍宝数量

分别为0,1,0。
第三年：在3号地区视察，2号地区会供奉一件珍宝到3号地区。此时每个地区的珍宝数量分别为0,0,1。

【样例3输入】

```
10
2 9 1 2 0 4 6 7 1 5
```

【样例3输出】

```
0 2 0 0 0 4 7 10 4 10
```

【数据范围】

对于30%的数据，保证$1 \leqslant n \leqslant 100$，$0 \leqslant a_i \leqslant 100$。
对于100%的数据，保证$1 \leqslant n \leqslant 5 \times 10^5$，$0 \leqslant a_i \leqslant 10^5$。

事件通报（report）

【题目描述】

小瓜在公司里面当秘书，现在有 n 条消息要告知老板。每条消息有一个好坏度，这会影响老板的心情。告知完一条消息后，老板的心情等于老板之前的心情加上这条消息的好坏度。最开始老板的心情是 0，一旦老板心情到了 0 以下就会勃然大怒，炒了小瓜的鱿鱼。

小瓜为了不被炒，提前知道每个消息的好坏度（已经按时间的发生顺序进行了排列）。他可以使用一种叫 “倒叙”的手法，来改变告知老板的顺序。具体地，如果有 n 条消息，小瓜可以任取一个整数 k（1≤k≤n），先从第 k 个事件通报到第 n 个事件，再从第一个事件通报到第 k-1 个事件。特别地，k=1 时按照原顺序通报。

他希望知道，有多少个这样的 k 可以让老板不发怒。

【输入格式】

从文件 report.in 中读入数据。

第一行：输入一个整数 n，表示消息总数。

第二行：输入 n 个整数 $a_1,a_2,...,a_n$，分别表示每条消息的好坏度。

【输出格式】

输出到文件 report.out 中。

输出一个整数，表示可行的方案数。

【样例 1 输入】

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

【样例 1 输出】

```
2
```

【样例 1 解释】

事件顺序的第一种方案：2→3→4→1，老板的好坏度变化为：5→1→2→(−3)，对应 k=2。

第二种方案：5→1→2→(−3) ，老板的好坏度变化为：1→2→(−3)→5，对应 k=3。

因此共有 2 种方案。

【样例 2 输入】

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

【样例 2 输出】

```
3
```

【样例 2 解释】

k=1，k=4，以及 k=5 均为合法方案。

【数据范围】

对于 25%的数据，保证 $1 \leqslant n \leqslant 10^3$。
对于 75%的数据，保证 $1 \leqslant n \leqslant 10^4$。
对于 100%的数据，保证 $1 \leqslant n \leqslant 10^6$，$-10^3 \leqslant a_i \leqslant 10^3$。