

A 因吹斯听

时空限制：1s 64MB

文件名

interesting.in/interesting.out/interesting.cpp

题目描述

对于一个长度为 $2 \times K$ 的序列，如果它的前 K 个元素之和小于等于 S 且后 K 个之和也小于等于 S ，我们则称之为 interesting。现给定一个长度为 N 的序列 a ，要求输出以每个元素开头能找到的最长 interesting 序列的长度（选出的序列必须在序列 a 中连续）。

输入格式

第一行两个整数 N, S 。

接下来 N 行，每行一个正整数，第 i 行表示序列中的第 i 个元素 a_i 。

输出格式

输出共 N 行，每行一个整数，第 i 行表示以 a_i 开头的最长的 interesting 序列。如果不存在，则输出 0。

样例 #1

样例输入 #1

```
5 10000
1
1
1
1
1
```

样例输出 #1

```
4
4
2
2
0
```

样例 #2

样例输入 #2

```
5 9
1
1
10
1
9
```

样例输出 #2

```
2
0
0
2
0
```

样例 #3

样例输入 #3

```
8 3
1
1
1
1
1
1
1
1
1
1
```

样例输出 #3

```
6
6
6
4
4
2
2
0
```

提示

【数据规模与约定】

对于 30% 的数据， $2 \leq N \leq 10^3$ 。

对于 100% 的数据， $2 \leq N \leq 10^5$ ， $1 \leq S, a_i \leq 2 \times 10^9$ 。

B 分赃

时空限制：2s 512MB

文件名

kas.in/kas.out/kas.cpp

题目描述

Kile 和 Pogi 在路上捡到了 N 张钞票。他们将分别取走若干张钞票，使得每人所得的总金额相同。同时要尽可能保证分得的总金额最大。

接着，他们会带着剩下的钞票前往赌场。由于他们的运气很好，因此他们将剩余金额作为赌注后，会得到双倍的金
额。然后，他们会将得到的金额再次平分，并加入每个人的总金额中。

求每个人能够分得的总金额是多少。

输入格式

第一行，一个整数 N 。

接下来的 N 行，每行一个正整数 c_i ，表示第 i 张钞票的面额。保证 N 张钞票的总金额不超过 10^5 。

输出格式

输出每个人能够分得的总金额。

样例 #1

样例输入 #1

```
4
2
3
1
6
```

样例输出 #1

```
6
```

样例 #2

样例输入 #2

```
5
2
3
5
8
13
```

样例输出 #2

```
18
```

提示

【样例 1 解释】

Kile 可以选择取走面额分别为 2, 3, 1 的钞票，而 Pogi 可以取走面额为 6 的钞票。由于没有剩余钞票，因此每人所得总金额为 6。

【样例 2 解释】

Kile 可以选择取走面额分别为 5, 8 的钞票，而 Pogi 可以取走面额为 13 的钞票。剩下的钞票面额分别为 2, 3，因此在前往赌场之后，每人所得总金额为 $13 + 2 + 3 = 18$ 。

【数据规模与约定】

对于 50% 的数据， $N \leq 13$ 。

对于 70% 的数据， $N \leq 50$ ， $\sum c_i \leq 1000$ 。

对于 100% 的数据， $1 \leq N \leq 500$ 。

C 网球

时空限制：1s 512MB

文件名

tenis.in/tenis.out/tenis.cpp

题目描述

你是网球比赛的组织者，一场有 n 个选手的比赛就要举办了，选手编号为 1 到 n 。在赛前你测试了他们在三种场地（A 场地，B 场地，C 场地）分别的初始排名，在这场比赛中，两个人之间比赛的场数恰好为一次，也就是说总共有 $\frac{n \times (n - 1)}{2}$ 场比赛。

比赛不会出现平局，当有平局情况出现时，会选择这次比赛所在的场地类型中初始排名靠前的选手获胜。你办的比赛也有一些黑幕，你就希望让你办的比赛中的获胜者能在他所更擅长的场地类型上比赛（即在他初试排名靠前的场地类型上比赛），如果有多个场地满足要求，那么选择输的那一方排名较靠前的。如果还有多个场地满足要求，选择编号小的（ $A < B < C$ ）。

你想知道，每个场地上举行了多少场比赛，且每个人赢的场数。

输入格式

第一行一个整数 n 代表选手个数。

第二行 n 个整数代表选手们在 A 场地的初始排名。

第三行 n 个整数代表选手们在 B 场地的初始排名。

第四行 n 个整数代表选手们在 C 场地的初始排名。

输出格式

第一行三个整数分别代表在 A 场地，B 场地，C 场地举办了多少场比赛。

第二行 n 个整数代表每个选手获胜了几场。

样例 #1

样例输入 #1

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

样例输出 #1

```
1 2 0
2 0 1
```

样例 #2

样例输入 #2

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

样例输出 #2

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

提示

样例 1 解释

- 第 1 名和第 2 名选手在 B 场地进行比赛，因为第 1 名选手在 B 场地的排名靠前。
- 第 2 名和第 3 名选手可以在 A 场地和 C 场地进行比赛，且输的那一方（第 1 名选手）排名也相同，那么选择编号小的 A 场地进行比赛。
- 第 1 名和第 3 名选手可以在 A,B,C 场地进行比赛，但是在 B 场地输的那一方（第 1 名选手）排名更靠前，所以选择 B 场地进行比赛。

【数据规模与约定】

对于 30% 的数据： $1 \leq n \leq 300$ 。

对于 40% 的数据，： $1 \leq n \leq 3000$ 。

对于 100% 的数据： $1 \leq n \leq 10^5$ 。

D 娃娃

时空限制：1s 64MB

文件名

vudu.in/vudu.out/vudu.cpp

题目描述

年轻的 Mirko 最近一直在买 Voodoo 娃娃。因为他对最便宜的东西很感兴趣，所以他每天都在追踪 Voodoo 娃娃的价格。他已经得知了最近 N 天的娃娃价格，第 i 天的娃娃价格记为 a_i 。

Mirko 认为，连续几天的娃娃平均价格与下一天的娃娃价格之间存在某种联系。他想验证自己的观点，却被一个问题难倒了：“对于一个给定的 P ，在这 N 天内有多少个不同的连续子序列令娃娃的平均价格大于或等于 P ？”

两个连续子序列不同当且仅当它们的开始位置或结束位置不同。

输入格式

第一行一个整数 N 。

接下来一行有 N 个整数，第 i 个整数表示 a_i 。

最后一行有一个整数 P 。

输出格式

一行一个整数，表示在这 N 天内有多少个不同的连续子序列令娃娃的平均价格大于或等于 P 。

样例 #1

样例输入 #1

```
3
1 2 3
3
```

样例输出 #1

```
1
```

样例 #2

样例输入 #2

```
3
1 3 2
2
```

样例输出 #2

```
5
```

样例 #3

样例输入 #3

```
3
1 3 2
3
```

样例输出 #3

```
1
```

提示

【样例 1 解释】

平均数大于等于 3 的子序列只有 3。

【样例 2 解释】

平均数大于等于 2 的子序列有 5 个，它们是：

1 3

1 3 2

3

3 2

2

【数据规模与约定】

对于 30% 的数据， $1 \leq N \leq 10^4$ ；

对于 100% 的数据， $1 \leq N \leq 10^6$ ， $1 \leq a_i \leq 10^9$ ， $1 \leq P \leq 10^9$ 。