

滚榜 (ranklist)

【题目描述】

封榜是 ICPC 系列竞赛中的一个特色机制。ICPC 竞赛是实时反馈提交结果的程序设计竞赛，参赛选手与场外观众可以通过排行榜实时查看每个参赛队伍的过题数与排名。竞赛的最后一小时会进行“封榜”，即排行榜上将隐藏最后一小时内的提交的结果。赛后通过滚榜环节将最后一小时的结果（即每只队伍最后一小时的过题数）公布。

Alice 围观了一场 ICPC 竞赛的滚榜环节。本次竞赛共有 n 支队伍参赛，队伍从 $1 \sim n$ 编号， i 号队伍在封榜前通过的题数为 a_i 。排行榜上队伍按照过题数从大到小进行排名，若两支队伍过题数相同，则编号小的队伍排名靠前。

滚榜时主办方以 b_i 不降的顺序依次公布了每支队伍在封榜后的过题数 b_i （最终该队伍总过题数为 $a_i + b_i$ ），并且每公布一支队伍的结果，排行榜上就会实时更新排名。Alice 并不记得队伍被公布的顺序，也不记得最终排行榜上的排名情况，只记得每次公布后，本次被公布结果的队伍都成为了新排行榜上的第一名，以及所有队伍在封榜后一共通过了 m 道题（即 $\sum_{i=1}^n b_i = m$ ）。

现在 Alice 想请你帮她算算，最终排行榜上队伍的排名情况可能有多少种。

【输入格式】

从文件 `ranklist.in` 中读入数据。

第一行包含两个正整数 n, m ，表示队伍数量与它们封榜后的总过题数。

第二行包含 n 个正整数表示 a_i 。

【输出格式】

输出到文件 `ranklist.out` 中。

仅一行一个整数表示答案。

【样例 1 输入】

```
1 3 6
2 1 2 1
```

【样例 1 输出】

```
1 5
```

【样例 1 解释】

- 1. 最终排名：1, 3, 2，滚榜情况（按公布顺序，下同）： $b_2 = 0, b_3 = 2, b_1 = 4$ 。
- 2. 最终排名：2, 1, 3，滚榜情况： $b_3 = 2, b_1 = 2, b_2 = 2$ 。
- 3. 最终排名：2, 3, 1，滚榜情况： $b_1 = 1, b_3 = 2, b_2 = 3$ 。
- 4. 最终排名：3, 1, 2，滚榜情况： $b_2 = 0, b_1 = 2, b_3 = 4$ 。
- 5. 最终排名：3, 2, 1，滚榜情况： $b_1 = 1, b_2 = 1, b_3 = 4$ 。

【样例 2 输入】

```
1 6 50
2 4 7 9 3 0 3
```

【样例 2 输出】

```
1 96
```

【样例 3 输入】

```
1 11 300
2 6 8 8 12 0 11 6 1 0 15 5
```

【样例 3 输出】

```
1 30140983
```

【数据范围】

对于所有测试数据： $1 \leq n \leq 13, 1 \leq m \leq 500, 0 \leq a_i \leq 10^4$ 。
每个测试点的具体限制见下表：

测试点编号	$n \leq$	$m \leq$
1 ~ 2	2	10
3 ~ 5	3	
6 ~ 8	8	100
9 ~ 12	10	200
13 ~ 16	12	300
17 ~ 20	13	500