pinely

Qaaxaap | 退出登录

You have +351! Wow!

Codeforces Better! 设置

主页 热门 指南目录 比赛 训练营 题库 团体 排行榜 培训 API 日历 帮助 RAYAN

问题 提交代码 我的提交 状态 黑客 房间 榜单 自定义调试

未找到 洛谷 未找到

C. Loyalty

time limit per test: 2 seconds

memory limit per test: 512 megabytes

No Loyalty No Royalty (c)

You are a customer in a store and want to buy n items. Each item i has a price a_i such that $1 \le a_i \le X$, where X is a **loyalty factor**.

Your **loyalty level** in the store is defined as $\lfloor \frac{S}{X} \rfloor$, where S is the total cost of items purchased so far. Initially, $S = 0$.

If you buy an item with price p and your loyalty level increases as a result of this purchase, you earn p bonus points.

Your task is to find the **maximum number of bonus points** you can earn by choosing an optimal order of purchase for the items.

有道 翻译

没有忠诚就没有版税(c)

您是一家商店的顾客，想购买 n 商品。每个商品 i 的价格 a_i 为 $1 \le a_i \le X$ ，其中 X 是一个**忠诚度因子**。

Pinely Round 5 (Div. 1 + Div. 2)

比赛进行中

02:51:52

Contestant

→ 提交?

语言: GNU G++17 7.3.0

选择文件: 选择文件 未选择文件

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

提交

→ 评分表

	Score
Problem A	491

您在商店中的**忠诚度级别**定义为 $\lfloor \frac{S}{X} \rfloor$ ，其中 S 是到目前为止购买的商品的总成本。最初，$S = 0$ 。

如果您购买价格为 p 的商品，并且您的忠诚度等级因此而提高，您将获得 p 奖励积分。

你的任务是找到*************你可以通过选择一个最优的购买顺序来赚取的奖励点数。

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 2 \cdot 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n ($1 \le n \le 10^5$) and X ($1 \le X \le 10^9$) — the number of items and the loyalty factor.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le X$) — the prices of the items.

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

有道 翻译

输入** **

每个测试包含多个测试用例。第一行包含测试用例的数量 t （ $1 \le t \le 2 \cdot 10^4$ ）。下面是测试用例的描述。

每个测试用例的第一行包含两个整数 n （ $1 \le n \le 10^5$ ） 和 X （ $1 \le X \le 10^9$ ） ——项目数和忠诚度因子。

每个测试用例的第二行包含 n 整数 a_1 、 a_2 、 ... 、 a_n （ $1 \le a_i \le X$ ） ——商品的价格。

保证所有测试用例 n 的和不超过 10^5 。

Output

For each test case, output two lines.

The first line should contain a single integer — the maximum number of bonus points that can be earned.

The second line should contain n integers — the prices of the items in an order of purchase that maximizes the number of bonus points.

Problem B	982
Problem C	1718
Problem D	2208
Problem E	2944
Problem F	3190
Problem G	3680
Problem H	4907
Successful hack	100
Unsuccessful hack	-50
Unsuccessful submission	-50
Resubmission	-50

* If you solve problem on 00:07 from the first attempt

If there are several orders that maximize the number of bonus points, you can output any of them.

有道 翻译

** **输出

对于每个测试用例，输出两行。

第一行应该包含一个整数-可以获得的最大奖励点数。

第二行应该包含 n 整数—购买订单中使积分最大化的商品价格。

如果有几个订单可以最大化奖励点数，你可以输出其中的任何一个。

Example

input Copy

```
7
10 2
1 2 1 2 1 2 1 2 1 2
5 10
2 2 2 2 5
11 23
5 5 22 1 21 2 10 3 1 1 2
1 1
1
1 17
11
3 100
44 32 1
16 100500
42801 73112 95296 68791 42217 21871 29316 84405 24273 42894 63370 53473 57156 61369 80 27290
```

output Copy

```
12
1 2 2 2 2 2 1 1 1 1
5
2 2 2 2 5
53
1 1 5 2 1 2 5 3 10 21 22
1
1
0
11
```

```
0
44 32 1
503499
53473 42894 80 57156 42801 61369 42217 63370 29316 68791 27290 73112 24273 84405 21871 95296
```

Note

In the first test case:

1. After buying the first item $S = 1$, loyalty level is 0;
2. After buying the second item $S = 3$, this increases loyalty level to 1 and earns 2 bonus points;
3. After buying the third item $S = 5$, this increases loyalty level to 2 and earns 2 bonus points;
4. After buying the fourth item $S = 7$, this increases loyalty level to 3 and earns 2 bonus points;
5. After buying the fifth item $S = 9$, this increases loyalty level to 4 and earns 2 bonus points;
6. After buying the sixth item $S = 11$, this increases loyalty level to 5 and earns 2 bonus points;
7. After buying the seventh item $S = 12$, this increases loyalty level to 6 and earns 1 bonus point;
8. After buying the eighth item $S = 13$;
9. After buying the ninth item $S = 14$, this increases loyalty level to 7 and earns 1 bonus point;
10. After buying the tenth item $S = 15$.

Overall we got 12 bonus points.

In the second test case:

1. After buying the first four items $S = 8$, loyalty level is 0;
2. After buying the last item $S = 13$, this increases loyalty level to 1 and earns 5 bonus points.

In the third test case:

1. After buying the first eight items $S = 20$, loyalty level is 0;
2. After buying the ninth item $S = 30$, this increases loyalty level to 1 and earns 10 bonus points;
3. After buying the tenth item $S = 51$, this increases loyalty level to 2 and earns 21 bonus points;
4. After buying the eleventh item $S = 73$, this increases loyalty level to 3 and earns 22 bonus points.

有道 翻译

注意

在第一个测试用例中：

1. 购买第一件商品 $S=1$ 后，忠诚度等级为0；
2. 购买第二件物品 $S=3$ 后，忠诚度等级将提升至 1 并获得 2 积分；
3. 购买第三件物品 $S=5$ 后，忠诚度等级将提升至 2 并获得 2 积分；
4. 购买第四件物品 $S=7$ 后，忠诚度等级将提升至 3 并获得 2 积分奖励；
5. 购买第五件物品 $S=9$ 后，忠诚度等级将提升至 4 并获得 2 积分；
6. 购买第六件物品 $S=11$ 后，忠诚度等级提升至 5 并获得 2 积分；
7. 购买第七件物品 $S=12$ 后，忠诚度等级提升至 6 并获得 1 积分奖励；
8. 购买第八项 $S=13$ 后；
9. 购买第九件物品 $S=14$ 后，忠诚度等级提升至 7 并获得 1 积分；
10. 在购买第十项 $S=15$ 之后。

总的来说，我们得到了 12 的加分。

在第二个测试用例中：

1. 购买前四件商品 $S=8$ 后，忠诚度等级为0；
2. 购买最后一件物品 $S=13$ 后，忠诚度等级将提升至 1 ，并获得 5 积分奖励。

在第三个测试用例中：

1. 购买前8件 $S=20$ 后，忠诚度等级为0；
2. 在购买第9件物品 $S=30$ 后，忠诚度等级将提升至 1 并获得 10 积分；
3. 购买第十件物品 $S=51$ 后，忠诚度等级提升至 2 ，并获得 21 积分奖励；
4. 购买第11件物品 $S=73$ 后，忠诚度等级将提升至 3 并获得 22 积分。

GNU G++17 7.3.0

▶ 自定义测试数据(自动保存)

▷

Codeforces (c) Copyright 2010-2025 Mike Mirzayanov

The only programming contests Web 2.0 platform
Server time: Oct/31/2025 00:42:56$^{\text{UTC+8}}$ (k3).
Desktop version, switch to mobile version.
Privacy Policy | Terms and Conditions

Supported by

ITMO