D. Bags and Coins

When you were a child you must have been told a puzzle of bags and coins. Anyway, here's one of its versions:

A horse has three bags. The first bag has one coin, the second bag has one coin and the third bag has three coins. In total, the horse has three coins in the bags. How is that possible?

The answer is quite simple. The third bag contains a coin and two other bags.

This problem is a generalization of the childhood puzzle. You have n bags. You know that the first bag contains a_1 coins, the second bag contains a_2 coins, ..., the n-th bag contains a_n coins. In total, there are s coins. Find the way to arrange the bags and coins so that they match the described scenario or else state that it is impossible to do.

Input

The first line contains two integers n and s $(1 \le n, s \le 70000)$ — the number of bags and the total number of coins. The next line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 70000)$, where a_i shows the number of coins in the i-th bag.

Output

If the answer doesn't exist, print -1.

Otherwise, print n lines, on the i-th line print the contents of the i-th bag. The first number in the line, c_i $(0 \le c_i \le a_i)$, must represent the number of coins lying directly in the i-th bag (the coins in the bags that are in the i-th bag are not taken into consideration). The second number in the line, k_i $(0 \le k_i < n)$ must represent the number of bags that lie directly in the i-th bag (the bags that are inside the bags lying in the i-th bag are not taken into consideration). Next, the line must contain k_i integers — the numbers of the bags that are lying directly in the i-th bag.

The total number of coins in the solution must equal s. If we count the total number of coins the i-th bag in the solution has, we should get a_i.

No bag can directly lie in more than one bag. The bags can be nested in more than one level (see the second test case). If there are multiple correct answers, you can print any of them.

Examples

input
```
3 3
1 3 1
```
output
```
1 0
1 2 3 1
1 0
```

input
```
3 3
1 3 1
```
output
```
1 0
2 1 3
0 1 1
```

input

```
1 2
1
```

output

```
-1
```

input

```
8 10
2 7 3 4 1 3 1 2
```

output

```
2 0
1 2 1 4
0 2 7 8
0 2 5 6
1 0
3 0
1 0
2 0
```

Note

The pictures below show two possible ways to solve one test case from the statement. The left picture corresponds to the first test case, the right picture corresponds to the second one.