



Problem 4 Safe Packing

The manager of a packing warehouse, which specialises in packing breakable items, contracted a supplier for boxes of sizes from the Fibonacci sequence. I am not sure of the reason behind it, but it is rumoured to be related to the recent “Da Vinci code” movie. An item whose size is in the sequence can be packed in a box of equal size with no filling, but an item whose size is not in the sequence must be packed with sufficient filling material to fill the box and protect the item from breaking. An item cannot be split between two boxes, and each item must be packed separately in its own box.

The second twist in this story, which makes it the more bizarre, is that the company only receives a daily filling material delivery of size F to be used. At the end of each day, any unused filling material is discarded.

Your task is to maximize the number of items shipped each day for a given amount of filling material F and a given list of items.

Fibonacci Numbers are defined as:

$$\begin{aligned} F(n) &= n && \text{for } n < 2 \\ &= F(n-1) + F(n-2) && \text{for } n > 1 \end{aligned}$$

A list of the first few Fibonacci numbers are:

0 1 1 2 3 5 8 13 21 34 55

Note that each number, with the exception of the first two, is obtained by adding the preceding two numbers.

INPUT:

Input to this problem consists of packing tasks for one or more days. The tasks for each day are described by two lines as follows:

- The first line consists of three integers: the number of items W , $0 < W < 1000$ to be packed; the available size of filling material, F , $1 < F < 1000$; and maximum size of each item, S , $1 < S < 100000000$. The integers are separated by a single space.
- The following line contains W integers separated by a single space that describe the sizes of items to be packed.

The input will be terminated by a line that consists of three zeros (0 0 0), separated by a single space. This line should not be processed.

OUTPUT:

For each day, the output consists of one line that contains the number of items that can be packed for that day.

EXAMPLE INPUT:

```
4 10 30
7 15 30 5
11 100 5812167
20 40 30 15 17 5812167 23 43 33 13 37
0 0 0
```

EXAMPLE OUTPUT:

```
3
10
```