LESS MONEY, MORE PROBLEMS

CHALLENGE DESCRIPTION:

Up until today, the nation you live in has used **D** different positive integer denominations of coin for all transactions. Today, the queen got angry when a subject tried to pay his taxes with a giant sack of low-valued coins, and she just decreed that no more than **C** coins of any one denomination may be used in any one purchase. For instance, if C = 2 and the existing denominations are 1 and 5, it is possible to buy something of value 11 by using two 5s and one 1, or something of value 12 by using two 5s and two 1s, but it is impossible to buy something of value 9 or 17.

You cannot directly challenge the queen's decree, but you happen to be in charge of the mint, and you can issue new denominations of coin. You want to make it possible for any item of positive value at most **V** to be purchased under the queen's new rules. (Note that this may not necessarily have been possible before the queen's decree.) Moreover, you want to introduce as few new denominations as possible, and your final combined set of pre-existing and new denominations may not have any repeats.

What is the smallest number of new denominations required?

INPUT SAMPLE:

The first argument is a path to a file. Each line of the input file contains one test case. Each test case consists of three parts separated by pipe symbol: 1) an integer **C**; 2) an integer **V**; 3) sorted list of space separated integers - all current denominations.

```
1 | 3 | 1 2
1 | 6 | 1 2 5
2 | 3 | 3
1 | 100 | 1 5 10 25 50 100
```

OUTPUT SAMPLE:

For each test case, output one line containing the minimum number of new denominations required, as described above.

```
0
1
1
3
```

CONSTRAINTS:

1. The number of test cases is 40.
2. 1 ≤ C ≤ 100.
3. $1 \le V \le 10^9$.
4. The number of current denominations is in range from 1 to 100.