7. Hard Money

Program Name: HardMoney.java **Input File: hardmoney.dat**

Ecco, the puzzler, is trying to figure out how many different ways he can make change for a given amount if he has different denominations of coins. For example current United States coins have denominations worth 1, 5, 10, 25, 50, and 100 pennies. Given these denominations there are various ways to make change for a given amount assuming you have unlimited numbers of coins in each denomination. For example with the current denominations there are 4 ways to have 11 cents. They are 11 pennies, 6 pennies and 1 nickel (worth 5 pennies), 1 penny and 2 nickels, and 1 penny and 1 dime.

But what if the denominations of coins were different? For example, what if the denominations of coins were 1, 3, 11, 17, and 50. In this case there would be 5 ways to have 11 cents. They are 11 pennies, 8 pennies and 1 three cent coin, 5 pennies and 2 three cent coins, 2 pennies and 3 three cent coins, and 1 eleven cent coin.

Write a program that, given the denominations of coins and the goal amount, prints the number of combinations of coins, using the given denominations, that add up to the goal amount. The goal amount and all denominations will be stated in units of pennies.

Input

- The first line will contain a single integer `n` that indicates the number of data sets that follow.
- Each data set will consist of 2 lines.
- The first line in a data set will contain two integers: `g d`.
    - `g` indicates the goal amount, where `g` will be greater than 0 and less than or equal to 10,000.
    - `d` indicates the number of different denominations of coins for this data set.
- The second line in a data set will contain `d` integers in ascending order separated by spaces. These are the denominations of coins for this data set. All denominations will be greater than 0 and less or equal to 1000. No denomination will appear more than once per data set.

Output

For each data set print the number of ways the given denominations can be combined to achieve the goal amount. Assume you have an unlimited number of coins of each denomination.

Example Input File

```
6
11 6
1 5 10 25 50 100
11 5
1 3 11 17 50
100 8
1 2 3 5 7 11 25 50
11 4
2 6 10 25
15 6
2 3 5 7 11 13
100 6
1 5 10 25 50 100
```

Example Output to Screen

```
4
5
98664
0
12
293
```