Prime days

Obviously, a day is *prime* if the day of the month is prime - in our calendar that would mean one of 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 or 31 (the latter two only being possible in some months). A day is *really prime* if it is prime and occurs in a prime-numbered month (i.e., February, March, May, July or November). But, a day is only *truly prime* if it is really prime *and* it is a prime-numbered day of the year. For example February 2 is really prime but not truly prime because it is the 33rd day of the year and 33 is not prime. In a non leap-year March 2 is truly prime because it's the 61st day of the year.

Problem Statement

In preparation for the launch of a universal calendar app (suitable for use on all planets and in all solar systems) you have been asked to develop a program that computes all the truly prime days of a year given only the lengths of the months. You may assume that the length of a year is never greater than 2 billion days.

Task

Write a program that takes input from the command-line a single sequence of month lengths (always positive integers) separated by spaces and prints to `stdout` all the truly prime days of the corresponding calendar, one per line in the following format:

```
<number of day>: <number of month> <day of month>
```

The output should be in order of `<number of day>`. The total number of days in the year (i.e., the sum of the command-line arguments) will not be greater than 2 billion.

For instance, if you submitted a java program called `App` in a package called `calendar` then the following input:

```
> java calendar.App 5 5 10
```

would produce the output:

```
7: 2 2
13: 3 3
17: 3 7
```

(points, Individual)