Problem 11: Combinations

20 minutes, 200 points
Filename: prob11 *(e.g. prob11.c, prob11.cpp, prob11.java, prob11.py2, prob11.py3)*

Description

Vendor provides packages of 7, 11, and 13 items. The cost to prepare each package is about the same. Determine the fewest packages required to fulfil an order for a specific number of items.

For example, 42 items can be packed into 1 package of 7, 2 packages of 11, and 1 package of 13, for a total of 4 packages (42 1 2 1 4).

Each line of input contains an integer representing the number of items ordered. Each input value will be a positive integer less than 1000. The last line of input will contain a zero. No other lines will contain a zero. When you receive a zero value, you can stop processing the input.

Each line of output should contain the original number of items, followed by the number of packages of 7, the number of packages of 11, the number of packages of 13, and the total number of packages. If the order cannot be fulfilled from any combination of packages, return the original number of items ordered with no other data (e.g. 27).

Sample Input

```
42
55
27
88
0
```

Sample Output

```
42 1 2 1 4
55 1 2 2 5
27
88 2 2 4 8
```

Hints

A simple (but inefficient) “brute force” approach to this problem would be to loop through each possible combination of packages, trying to fill the largest packages first, and returning the first combination of packages that contain the exact number of items ordered. The scoring system will accept a program that uses this approach, as long as your program does not run for more than 5 seconds on the scoring server.

Prepared by Jason Klein. Adapted from CodeWars