CCC 1996 - reconstructed statement

This PDF was reconstructed by QOJ from the archived official HTML. It is not an original CCC PDF. The unchanged original HTML is available as an attachment.

Problem A: Deficient, Perfect, and Abundant

Input file: `dpa.in`
Output file: `dpa.out`

Write a program that repeatedly reads a positive integer, determines if the integer is *deficient*, *perfect*, or *abundant*, and outputs the number along with its classification.

A positive integer, *n*, is said to be *perfect* if the sum of its proper divisors equals the number itself. (Proper divisors include 1 but not the number itself.) If this sum is less that *n*, the number is *deficient*, and if the sum is greater than *n*, the number is *abundant*.

The input starts with the number of integers that follow. For each of the following integers, your program should output the classification, as given below. You may assume that the input integers are greater than 1 and less than 32500.

Sample input

```
3
4
6
12
```

Sample output

```
4 is a deficient number.

6 is a perfect number.

12 is an abundant number.
```

QOJ adaptation notes

This is a faithful typeset reconstruction of the archived official HTML, not an original contest PDF. The original HTML is attached without changes.

Use standard input and standard output instead of the named input and output files. Blank-line differences are ignored; output tokens must otherwise follow the statement.

QOJ reconstructed scoring: the original independent test files are weighted equally for a total of 15 points. These weights are not claimed to be the historical CCC marking scheme. Each file is an all-or-nothing scoring group.