

# Abundant, Deficient, and Perfect Numbers

Problem ID: a07p05abundantdeficientperfect

Recall that a divisor  $a$  of an integer  $n$  is an integer that divides  $n$  with no remainder, such that the result of the division  $n/a$  is an integer.

Formally, let  $a, n$  be integers. If there exists an integer  $b$  such that  $n = a \cdot b$ , then  $a$  is called a divisor of  $n$ , also called a factor of  $n$ . In other words, we say  $a$  divides  $n$ , that  $n$  is divisible by  $a$  and that  $n$  is a multiple of  $a$ . Of course, the same goes for  $b$  as for  $a$ .

A number is abundant if the sum of its divisors, other than itself, is larger than the number itself, deficient if the sum of divisors is lower, and perfect if the sum of divisors equals the number. For example, 6 is a perfect number because the divisors of 6 are 1, 2, 3, 6 and the sum of its divisors, other than 6 itself, is  $1 + 2 + 3 = 6$ .

Write the function `sum_of_divisors(number)`, which takes as an argument an integer  $n$  and sums all its divisors, excluding itself. For example, `sum_of_divisors(12)` returns 16 because the divisors of 12 are 1, 2, 3, 4, 6, 12, and  $1 + 2 + 3 + 4 + 6 = 16$ . Note that we exclude 12 itself in this sum.

Now, write a second function `decide(number)`. This function receives as a parameter an integer  $n$ , and returns one of the following strings:

- “{n} is abundant.” if the sum of the divisors of  $n$  are greater than  $n$  itself.
- “{n} is deficient.” if the sum of the divisors of  $n$  is less than  $n$  itself.
- “{n} is a perfect number.” if the sum of the divisors of  $n$  is equal to  $n$  itself.

**Note that we are testing your code differently in this task, please only submit your function definitions, without any code outside the functions!** The main python file, which handles input and output, is already provided. You can download and place the main file in the same directory as your python file. You can then run the main python file we provide to try out the samples.

Note that only the function `decide` will be tested, but you should make that function delegate some of the work it needs to perform, by calling the other function, `sum_of_divisors`.

## Input

The input to the function `decide` will consist of one integer  $n$ , the number to be checked, where  $1 \leq n \leq 10\,000$ . Only non-negative integers will be tested as input. As usual, input validation is not part of the program requirements. We'll get to that later.

## Output

The output of the function should be one string, containing the verdict as described above.

Sample Input 1	Sample Output 1
6	6 is a perfect number.
Sample Input 2	Sample Output 2
10	10 is deficient.
Sample Input 3	Sample Output 3
66	66 is abundant.