Count Abundant Numbers

In number theory, an abundant number is a number for which the sum of its proper divisors is greater than the number itself. Proper divisors of a number include all the divisors excluding the number itself. For example, the integer 12 is the first abundant number. Its proper divisors are 1, 2, 3, 4, and 6, which sum up to 16. This sum is greater than the number 12 itself. On the other hand, the integer 14 is not an abundant number as its proper divisors are 1, 2, and 7, which sum up to 10, not greater than 14.

You are tasked with writing a program that takes a positive integer n (greater than 1) as input and outputs the number of abundant numbers in the range from 2 to n, inclusive. If the input number is less than 2, the program should output "Invalid Input".

Function Description

Complete the `count_abundant_numbers` function in the editor below.

`count_abundant_numbers` has the following parameter:

- `int n`: the upper limit of the range to check for abundant numbers.

Returns

- `int`: the number of abundant numbers from 2 to n inclusive. If $n<2$, print "Invalid Input".

Input Format

A single integer *n*.

Constraints

- *n* is a positive integer.

Sample Input & Output

Input	Output
15	Number of abundant numbers from 1 to 15 is 1
1	Invalid Input

Explanation

- For the input 15, the only abundant number between 2 and 15 is 12. Hence, the output is 1.
- For the input 1, the input is invalid per the problem constraints, so the output is "Invalid Input".