Identify Smith Numbers

A *Smith number* is a composite number, the sum of whose digits is the sum of the digits of its prime factors obtained as a result of prime factorization (excluding 1). The first few such numbers are 4, 22, 27, 58, 85, 94, and 121.

Example:
$378 = 2 \times 3 \times 3 \times 3 \times 7$
So, its prime factors are 2, 3, 3, 3, and 7.
The sum of its digits is $(3 + 7 + 8) = 18$.
The sum of the digits of its factors is $(2 + 3 + 3 + 3 + 7) = 18$.

Similarly, 4937775 is a Smith number.
$4937775 = 3 \times 5 \times 5 \times 65837$, and the sum of its digits is the same as the sum of the digits of its prime factors: $4 + 9 + 3 + 7 + 7 + 7 + 5 = 3 + 5 + 5 + 6 + 5 + 8 + 3 + 7 = 42$.

Task:
Write a program to check whether a given integer is a Smith number.

Input Format

There will be only one line of input: N, the number which needs to be checked.

Constraints:
$0 < N < 2,147,483,647$ *(max value of an integer of the size of* 4 *bytes)*

Output Format

1 if the number is a Smith number.
0 if the number is a not Smith number.

Sample Input

378

Sample Output

1

Explanation

Its prime factors are 2, 3, 3, 3, and 7.
The sum of its digits is $(3 + 7 + 8) = 18$.
The sum of the digits of its factors is $(2 + 3 + 3 + 3 + 7) = 18$.