

# Sieve of Eratosthenes

Previously we have done an exercise about finding a number's factors, which can help us determine if the number is prime or not.

If we want to generate a list of primes, instead of checking the primality of each number, the Sieve of Eratosthenes is a more efficient way to do so. This is how it is performed:

- Let's say you want a list of all primes  $\leq N$ . Create a list of all integers from 2 to  $N$ .
- Repeat the following until the list is empty:
  - The smallest integer  $p$  in the list is prime.
  - Remove all multiples of  $p$  from the list.

Very often, the list is implemented using an array of boolean of size  $N$ . Initially the array is all true (except 0 and 1 which are false), meaning that the numbers exist in the list. To delete a number  $x$ , change the  $x^{\text{th}}$  element of the array to false. Please read the pseudo-code in the Wikipedia article for more details.

Write a program to generate a list of primes from 1 to  $N$ .

## INPUT

The only line contains an integer  $N$ . ( $2 \leq N \leq 10^7$ )

## OUTPUT

Output the primes in increasing order in separate lines.

## SAMPLE TESTS

|          | Input | Output   |
|----------|-------|--|
| <b>1</b> | 30    | 2<br>3<br>5<br>7<br>11<br>13<br>17<br>19<br>23<br>29 |