Algorithm 1: Sieve of Eratosthenes

```
   Input  : Integer X
   Output: List of Prime Numbers up to X or Error Message
 1 numberList = list from 2 to X;
 2 sqrtNumber = square root of X;
 3 indexSqrt = index of sqrtNumber in numberList;
 4 for i ← 0 to indexSqrt do
   |  /* optimization to stop at square root of X            */
 5 |  currentNumber = value at i in numberList;
 6 |  if currentNumber is not -1 then
 7 |  |  for each value V in numberList do
   |  |  |  /* iterate over all numbers in numberList       */
 8 |  |  |  if V is not -1 AND V is not currentNumber AND
   |  |  |   (V)%(CurrentNumber) is 0 then
 9 |  |  |  |  numberList at index of V = -1
10 |  |  |  else
11 |  |  |  |  do nothing
12 |  |  |  end
13 |  |  end
14 |  else
15 |  |  do nothing
16 |  end
17 end
18 remove all elements with value -1 from numberList;
```