```
clear
clc
% N changes from 2 to 30
N = 2:30;
```

using 'rem' : Eratosthenes sieve algorithm

initialize the (length(N) x 2) matrix 'primeNumbers' and assume all N as a prime at first (set true)

```
tic
primeNumbers = N';
primeNumbers(:, 2) = true;

% (1) if N has a prime as divisor, rem(N,prime) = 0
% (2) every prime has 1 and itself as divisors
% from (1) and (2), prime is the first N satisfying "rem(N, prime)==0"
% the others is not prime (set false)

% for all n < N, Nxn has already checked
% therefore Ns that satisfy "NxN < N(end)" are enough
for idx = find(N.^2 < N(end))
    if primeNumbers(idx, 2)
        multiplesOfNidx = find(rem(N, N(idx))==0);
        primeNumbers(multiplesOfNidx(2:end), 2) = false;
    end
end
display_res(primeNumbers);
```

```
2 is prime
3 is prime
4 is not prime
5 is prime
6 is not prime
7 is prime
8 is not prime
9 is not prime
10 is not prime
11 is prime
12 is not prime
13 is prime
14 is not prime
15 is not prime
16 is not prime
```

```
toc
```

```
경과 시간은 0.038782초입니다.
```

using 'isprime'

initialize the (length(N) x 2) matrix 'primeNumbers' and use 'isprime' to check whether N is prime

```
clear primeNumbers
tic
primeNumbers = N';
primeNumbers(:, 2) = isprime(N);
display_res(primeNumbers);
```

```
 2 is prime
 3 is prime
 4 is not prime
 5 is prime
 6 is not prime
 7 is prime
 8 is not prime
 9 is not prime
10 is not prime
11 is prime
12 is not prime
13 is prime
14 is not prime
15 is not prime
16 is not prime
```

```
toc
```

```
경과 시간은 0.043794초입니다.
```

display_res function

read a matrix row by row and display the result

```
function display_res(primeNumbers)
    for iRow = 1:size(primeNumbers, 1)
        if primeNumbers(iRow, 2)
            disp(int2str(primeNumbers(iRow, 1)) + " is prime");
        else
            disp(int2str(primeNumbers(iRow, 1)) + " is not prime");
        end
    end
    disp(primeNumbers); % show a matrix
end
```