see is_prime_number.mlx first

```
%parameter
K = 20;
```

built-in function

Notice: always check is there any built-in function first

```
tmp1 = primes(10000); %for large K, maybe should 10000 or even larger
prime_list = tmp1(1:K);
disp(prime_list)
```

```
     2     3     5     7    11    13    17    19    23    29    31    37    41    43    47    53    59    61    67
```

implement generate_K_prime yourself

```
my_prime_list = my_generate_K_prime(K);
disp(my_prime_list)
```

```
     2     3     5     7    11    13    17    19    23    29    31    37    41    43    47    53    59    61    67
```

function definition

```
function ret = my_generate_K_prime(K)
% K(int)
% ret(int,(1,K))
ret = zeros(1,K);
ret(1) = 2;
num_prime_number = 1;
current_x = 3;
while num_prime_number<K
    if my_isprime(current_x, ret(1:num_prime_number))
        num_prime_number = num_prime_number + 1;
        ret(num_prime_number) = current_x;
    end
    current_x = current_x + 2; %optimization 1
end
end

function ret = my_isprime(num1, known_prime)
% return true if num1 is prime number otherwise false
% num1(int)
% (ret)(bool)
for x = known_prime
```

```
    if x^2>num1, break; end%optimization 2
    if is_divisible(num1, x)
        ret = false;
        return
    end
end
ret = true;
end

function ret = is_divisible(a, b)
% return true if a is divisible by b otherwise false
% doc('mod')
ret = mod(a,b)==0;
end
```