

```

1  /*
2  * Scola Tutorial
3  * by Antoine B. Rauzy
4  * Eratosthenes' Sieve
5  */
6
7  block Numbers
8      integer maximum 100
9      integer candidate 3
10     symbol prime NONE
11     integer numberOfPrimes 0
12     integer index 0
13     block Primes
14         integer p1 2
15     end
16 end
17
18 scenario EratosthenesSieve as Numbers
19     state Initial
20     test IsDone
21         case yes (geq candidate maximum)
22         case no (lt candidate maximum)
23     end
24     task HandleCandidate
25         set numberOfPrimes (size 'Primes)
26         set index 0
27     end
28     test AreAllPrimesTested
29         case yes (eq index numberOfPrimes)
30         case no (lt index numberOfPrimes)
31     end
32     task GetPrime
33         set prime (element 'Primes index)
34     end
35     test isPrimeCandidateDivisor
36         case yes (eq (mod candidate (eval prime)) 0)
37         case no (df (mod candidate (eval prime)) 0)
38     end
39     task StoreNewPrime
40         new integer (append Primes (symbol (append "p" (string (add (eval
41             numberOfPrimes) 1)))) candidate
42     end
43     task IncrementIndex
44         set index (add index 1)
45     end
46     task GetNextCandidate
47         set candidate (add candidate 1)
48     end
49     state Final
50     next Initial IsDone
51     next IsDone.yes Final
52     next IsDone.no HandleCandidate
53     next HandleCandidate AreAllPrimesTested
54     next AreAllPrimesTested.yes StoreNewPrime
55     next AreAllPrimesTested.no GetPrime
56     next StoreNewPrime GetNextCandidate
57     next GetPrime isPrimeCandidateDivisor
58     next isPrimeCandidateDivisor.no IncrementIndex
59     next isPrimeCandidateDivisor.yes GetNextCandidate
60     next IncrementIndex AreAllPrimesTested
61     next GetNextCandidate IsDone
62 end

```