Sieve_Of_Eratosthenes

The sieve of Eratosthenes is one of the most efficient ways to find all primes smaller than n when n is smaller than 10 million.

Algorithm

Step 1.

Create a list of consecutive integers from 2 to n: (2, 3, 4, ..., n).

Step 2.

Initially, let p equal 2, the first prime number.

Step 3.

Starting from p^2, count up in increments of p and mark each of these numbers greater than or equal to p^2 itself in the list.

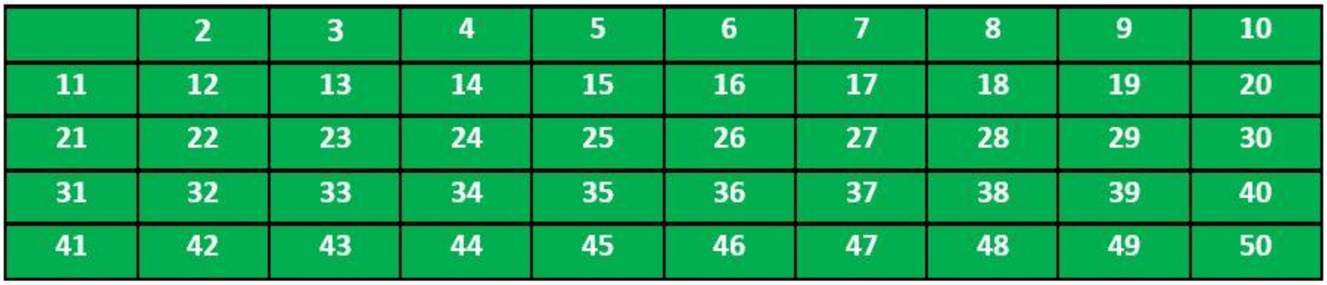
Step 4.

Find the first number greater than p in the list that is not marked. If there was no such number, stop. Otherwise, let p now equal this number (which is the next prime), and repeat from step 3.

When the algorithm terminates, all the numbers in the list that are not marked are prime.

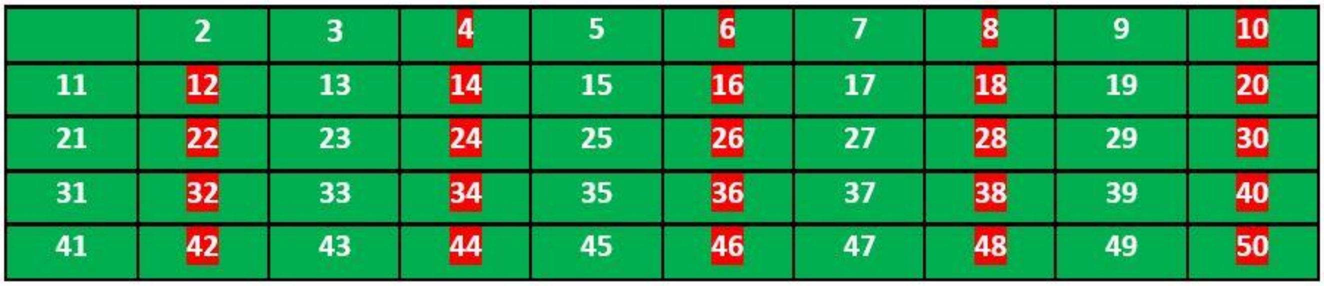
Example.(according to algorithm)

Step 1.



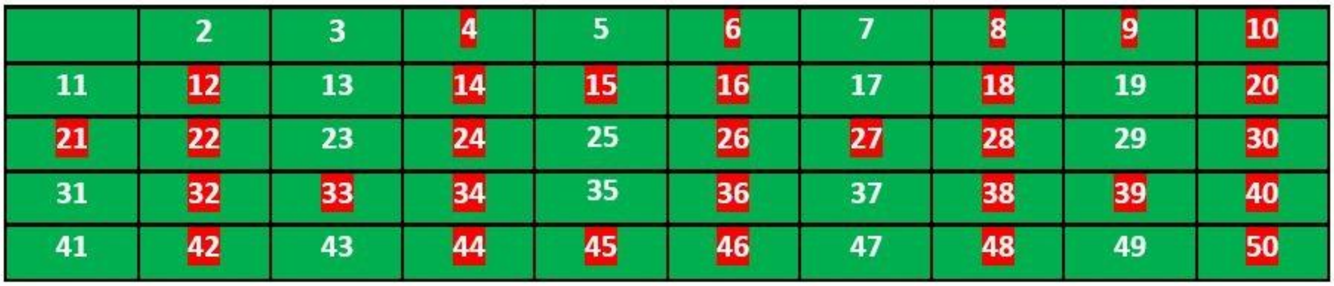
	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Step 2.



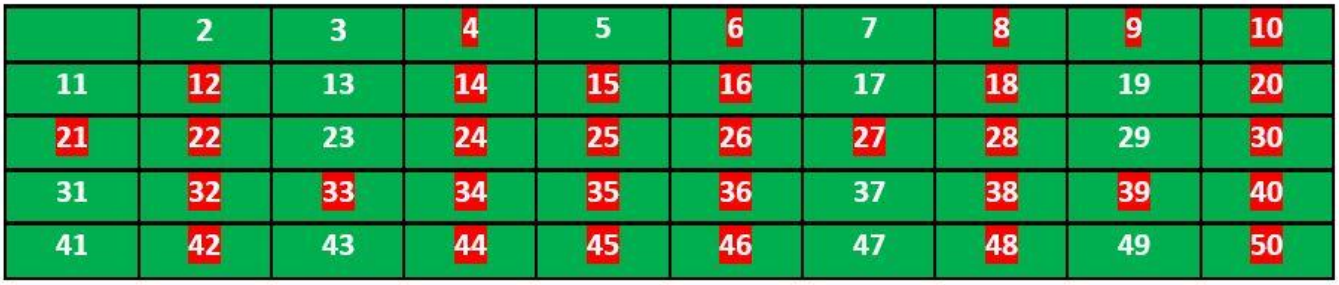
	2	3	**4**	5	**6**	7	**8**	9	**10**
11	**12**	13	**14**	15	**16**	17	**18**	19	**20**
21	**22**	23	**24**	25	**26**	27	**28**	29	**30**
31	**32**	33	**34**	35	**36**	37	**38**	39	**40**
41	**42**	43	**44**	45	**46**	47	**48**	49	**50**

Step 3.



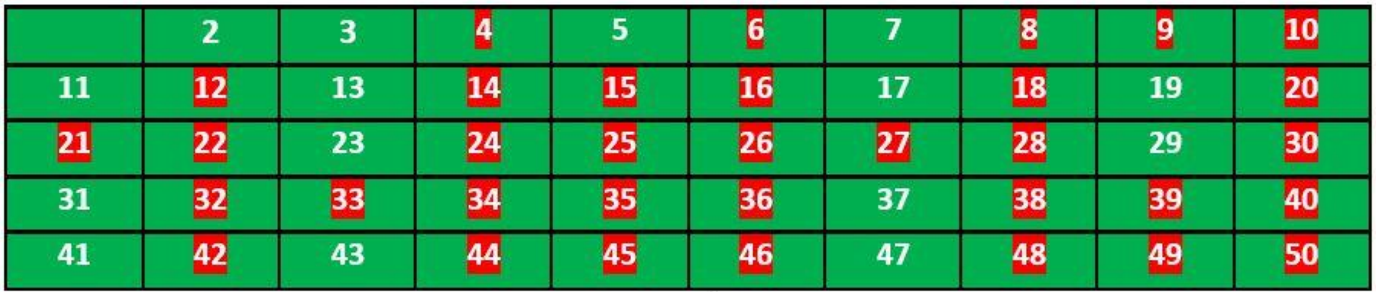
	2	3	**4**	5	**6**	7	**8**	**9**	**10**
11	**12**	13	**14**	**15**	**16**	17	**18**	19	**20**
21	**22**	23	**24**	25	**26**	**27**	**28**	29	**30**
31	**32**	**33**	**34**	35	**36**	37	**38**	**39**	**40**
41	**42**	43	**44**	**45**	**46**	47	**48**	49	**50**

Step 4.



	2	3	**4**	5	**6**	7	**8**	**9**	**10**
11	**12**	13	**14**	**15**	**16**	17	**18**	19	**20**
21	**22**	23	**24**	**25**	**26**	**27**	**28**	29	**30**
31	**32**	**33**	**34**	**35**	**36**	37	**38**	**39**	**40**
41	**42**	43	**44**	**45**	**46**	47	**48**	49	**50**

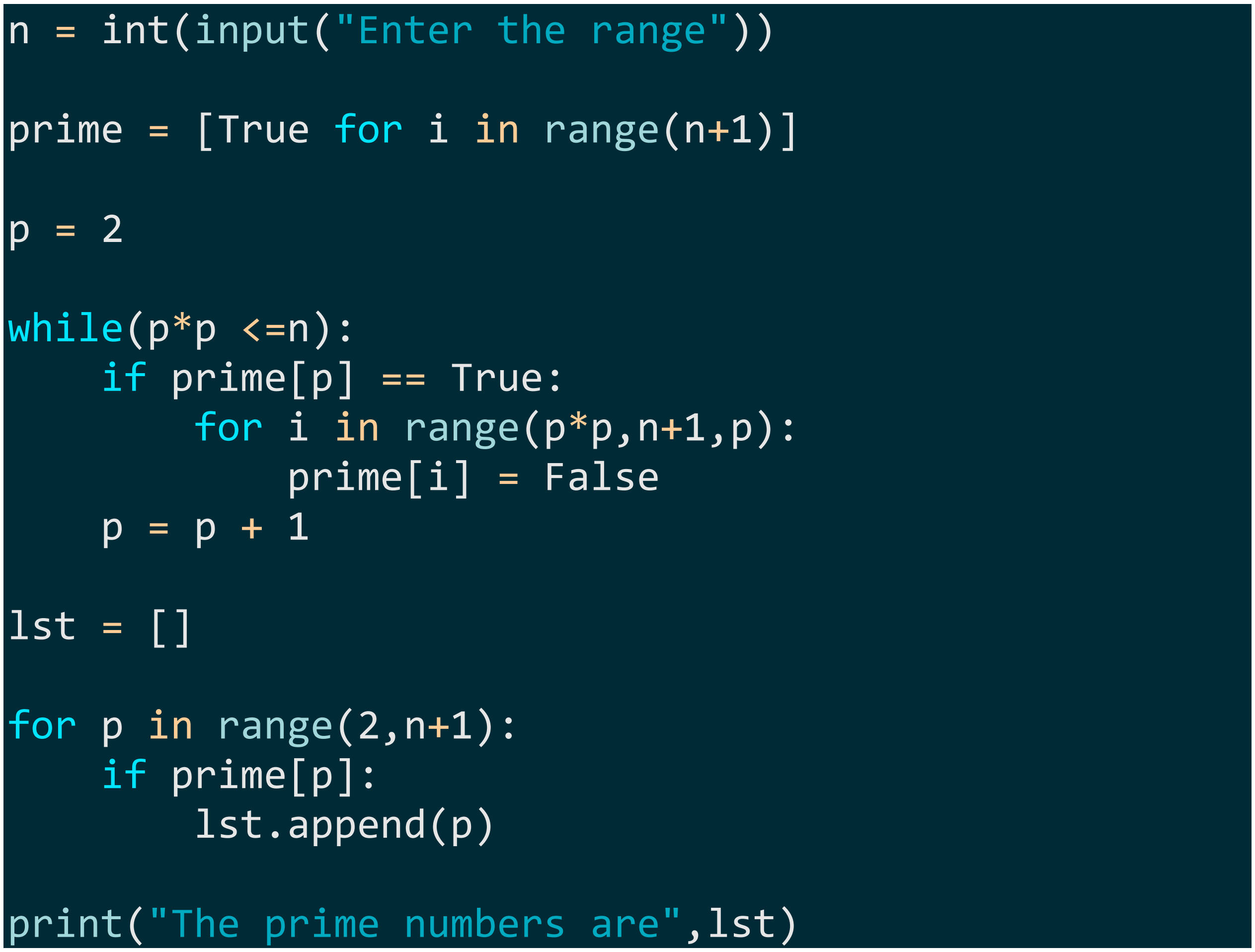
Step 5.



	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

So the prime numbers are the unmarked ones: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47.

Implementation:



```
n = int(input("Enter the range"))

prime = [True for i in range(n+1)]

p = 2

while(p*p <=n):
    if prime[p] == True:
        for i in range(p*p,n+1,p):
            prime[i] = False
    p = p + 1

lst = []

for p in range(2,n+1):
    if prime[p]:
        lst.append(p)

print("The prime numbers are",lst)
```