**What is Sieve of Eratosthenes ?**

1. Sieve of Eratosthenes is a simple and ingenious ancient algorithm for finding all prime numbers up to any given limit.

2. It does so by iteratively marking as composite (i.e., not prime) the multiples of each prime, starting with the first prime number, 2.

3. The multiples of a given prime are generated as a sequence of numbers starting from that prime, with constant difference between them that is equal to that prime.

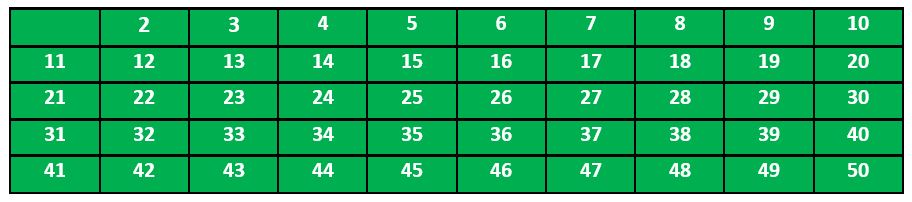
4. Following is the algorithm to find all the prime numbers less than or equal to a given integer n by Eratosthenes’ method :

* Create a list of consecutive integers from 2 to n : (2, 3, 4, ..., n).
* Initially, let p equal 2, the first prime number.
* 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. These numbers will be p(p+1), p(p+2), p(p+3), etc.
* 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 c.

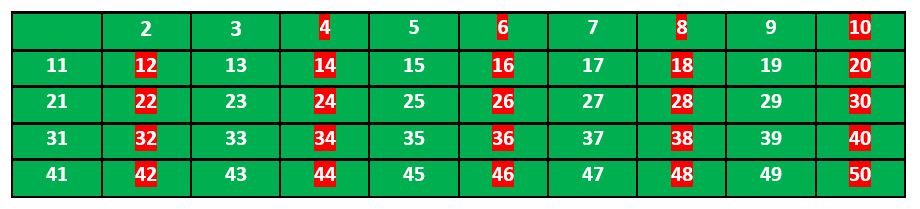
**Que 4.2. Explain Sieve of Eratosthenes with example.**

[view full article](https://www.geeksforgeeks.org/sieve-of-eratosthenes/)

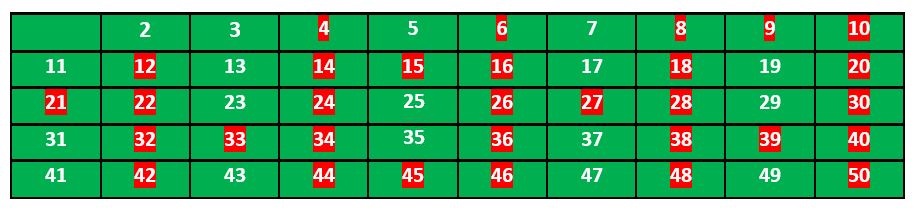
**Explanation with Example:**   
Let us take an example when n = 50. So we need to print all prime numbers smaller than or equal to 50.   
We create a list of all numbers from 2 to 50.



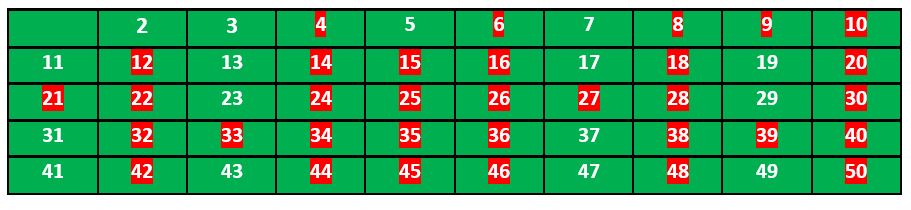
According to the algorithm we will mark all the numbers which are divisible by 2 and are greater than or equal to the square of it.



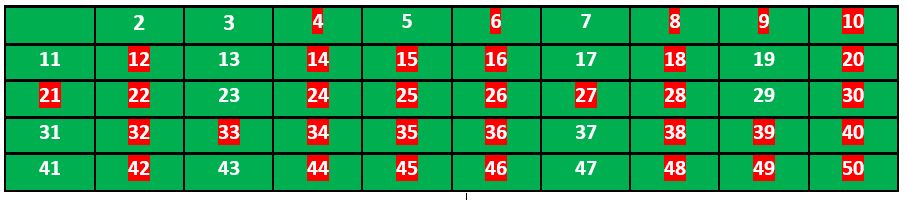
Now we move to our next unmarked number 3 and mark all the numbers which are multiples of 3 and are greater than or equal to the square of it.



We move to our next unmarked number 5 and mark all multiples of 5 and are greater than or equal to the square of it.



We continue this process and our final table will look like below:



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

**Que 4.3. Write a Python program to print all primes smaller than or equal to n using Sieve of Eratosthenes.**

def prime\_eratosthenes(n):

prime\_list = []

for i in range(2, n+1):

if i not in prime\_list:

print (i)

for j in range(i\*i, n+1, i):

prime\_list.append(j)

print(prime\_eratosthenes(100))

**Explain abstract data types with its types.**

Answer

1. Abstract Data type (ADT) is a type for objects whose behaviour is defined

by a set of value and a set of operations.

2. There are three types of ADTs :

**a. List ADT :** The data is stored in key sequence in a list which has a

head structure consisting of count, pointers and address of compare function needed to compare the data in the list.

**b. Stack ADT :**

i. In stack ADT Implementation instead of data being stored in each node, the pointer to data is stored.

ii. The program allocates memory for the data and address is passed to the stack ADT.

iii. The head node and the data nodes are encapsulated in the ADT.

iv. The calling function can only see the pointer to the stack.

v. The stack head structure also contains a pointer to top and count of number of entries currently in stack.

**c. Queue ADT :**

i. The queue abstract data type (ADT) follows the basic design of the stack abstract data type.

ii. Each node contains a void pointer to the data and the link pointer to the next element in the queue.

iii. The program allocates the memory for storing the data.

**Explain ADT interface in Python programming.**

Answer

1. ADT only defines as what operations are to be performed but not how

these operations will be implemented.

2. It does not specify how data will be organized in memory and what

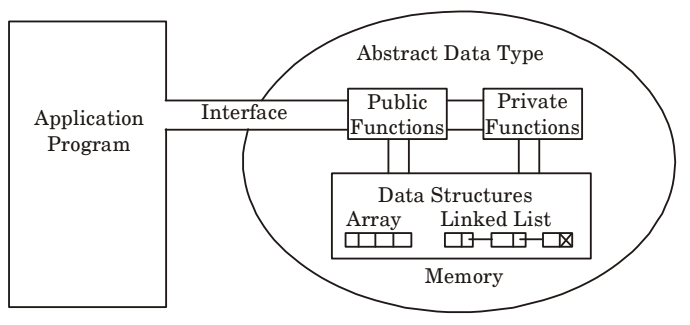
algorithms will be used for implementing the operations.

3. It is called “abstract” because it gives an implementation-independent

view.

4. The process of providing only the essentials and hiding the details is

known as abstraction.



5. The user of data type does not need to know how that data type is

implemented, for example, we have been using primitive values like int,

float, char data types only with the knowledge that these data type can

operate and be performed on without any idea of how they are implemented.

6. So, a user only needs to know what a data type can do, but not how it will

be implemented.

Code: <https://citizenchoice.in/course/Python-Programing/Unit%204/Modules-Import-Statements-and-Abstract-Data-Types>