Sieve of Eratosthenes

```
I/p: n = 10
O/p: 2 3 5 7
I/p: n = 23
O/p: 2 3 5 7 11 13 17 19 23
I/p: n = 7
O/p: 2 3 5 7
```

Efficient Solution

```
def SieveOfEratosthenes(num):
    prime = [True for i in range(num+1)]
    p = 2
    while (p * p <= num):
        if (prime[p] == True):
            for i in range(p * p, num+1, p):
                prime[i] = False
        p += 1
    for p in range(2, num+1):
        if prime[p]:
            print(p)


num = 50
print("Following are the prime numbers smaller"),
print("than or equal to", num)
SieveOfEratosthenes(num)
```

```
def isPrime(x):
    if n == 1:
        return False
    if n == 2 or n == 3:
        return True
    if n % 2 == 0 or n % 3 == 0:
        return False
    i = 5
    while (i * i <= n):
        if n % i == 0 or n % (i + 2) == 0:
            return False
        i += 6
    return True

def printPrimes(n):
    for i in range(2, n + 1):
        if isPrime(i):
            print(i, end=" ")

n = 18
printPrimes(n)
```

More Efficient Solution

```
def SieveOfEratosthenes(num):
    if n <= 1:
        return
    isPrime = [True] * (n + 1)
    i = 2
    while i <= n:
        if isPrime[i]:
            print(i, end=" ")
            for j in range(i * i, n + 1, i):
                isPrime[j] = False
        i += 1

num = 50
SieveOfEratosthenes(num)
```