

```
import timeit
# Eratosthenes Eleği Asal Sayıları
def eratosthenes_sieve(n):
    # 1. İlk adımda, 2'den n'ye kadar olan tüm sayıları içeren bir liste oluşturuyoruz.
    sayilar = list(range(2, n + 1))
    asal_sayilar = [] # Asal sayıları saklayacağımız boş bir liste oluşturuyoruz.

    while sayilar:
        # 2. Her adımda, listenin ilk elemanı her zaman bir asal sayıdır.
        asal = sayilar[0]
        asal_sayilar.append(asal) # Bu asal sayıyı "asal_sayilar" listesine ekliyoruz.

        # 3. Şimdi bu asal sayının katlarını listeden çıkaracağız.
        # İlk adımda, bu asal sayı 2'dir, bu yüzden 2'nin katlarını ele alacağız.
        # Bir sonraki adımda, bir sonraki asal sayıyı kullanacağız ve bu sayının katlarını ele alacağız.
        sayilar = [x for x in sayilar if x % asal != 0]

    # 4. Döngü bittiğinde, "sayilar" listesinde kalan tüm sayılar asal sayılardır.
    return asal_sayilar # Asal sayıları içeren "asal_sayilar" listesini döndürüyoruz.

# Kullanıcıdan bir üst sınır alalım
n = int(input("Asal sayıları bulmak için bir üst sınır girin: "))

# Eratosthenes Eleği'ni kullanarak asal sayıları bulalım.
asal_sayilar = eratosthenes_sieve(n)

# Sonuçları ekrana yazdıralım.
print("2'den", n, "kadar olan asal sayılar:")
print(asal_sayilar)

n = 1000 # Örnek olarak n = 1000 kullanalım.

# Hız ve performans ölçümü için timeit kullanıyoruz.
execution_time = timeit.timeit("eratosthenes_sieve(n)", globals=globals(), number=1000)
print(f"Eratosthenes Eleği ile {n} kadar olan asal sayıları bulma süresi: {execution_time:.6f} saniye")
```



```
Asal sayıları bulmak için bir üst sınır girin: 20
2'den 20 kadar olan asal sayılar:
[2, 3, 5, 7, 11, 13, 17, 19]
Eratosthenes Eleği ile 1000 kadar olan asal sayıları bulma süresi: 1.148311 saniye
```

