HW10

Dynamic Array Adımları :

1-)Data txt dosyasını okuma modunda açtım.
2-)Arrayime kendi belirlediğim hafıza boyutuna göre malloc ile bellekte yer ayırttım.
3-)Text dosyasından bir tane long int alınır ve while döngüsüne girilir.While döngüsü parametre olarak girilen finalNum sayısı kadar döner.Mesela finalNum 500000 ise data.txt dosyasından 500000 sayısına kadar long int alınır.
4-)Her alınan sayının asal olup olmadıgı kontrol edilir.[Asal Alma Fonksiyonum : eğer bir sayı asalsa 2 den onun karakköküne kadar olan sayılardan en az birine bölünür.Asal olan sayılar ise bu aralıktaki hiç bir sayıya bölünmez.
5-)Used adlı bir değişken belirledim bu değişken hem arrayin indexini hemde bellekte kullanılan yer sayısını tutacak.Used değişkeninin ilk değeri sıfırdır ve her asal olan sayıyı bulundulça bir artırılır.
6-)Eğer kullanılan arrayin (yani usedin) bir fazlası array için bellekten ayırtılmış yere denk geliyorsa bellek boyutu iki katına çıkarılır ve ek olarak tanımladığım array için bu bellek boyutu kadar yer alınır.Daha sonra ana arrayimin elemanlarını tutması için elemanları ikinci arraye atanır.Bellekte yer kaplamaması için ana arrayim için aldığım ilk yeri free ile belleğe geri veririm.Son olarak da ana arrayimin yerini ikinci olarak eklediğim arrayin yeri olarak gösteririm.
7-)Dosyanın sonundaki EOF un alınmaması için eğer dosyadan alınan bir satırda 3 tane eleman yoksa (yani %ld %c %d) number sayısı 10 artırılarak finalNum dan daha büyük olur ve döngüden çıkar.
8-)İşlemler bittiğinde data.txt dosyası kapatılır.
9-)Eğer parametre olarak girilen digit sayısı 1 ise dosyadaki asal sayılar parametre olarak gönderilen outputFile ın gösterdiği dosyaya yazdırılır.[Digit Num : parametre olarak gönderilen 1 ve ya 0 değerleridir.Bu değerler asal sayıların yazdırılıp yazdırılmayacağını belirler.1 sayısı yazdırılacağını,0 ise bastırlmayacağını söyler.
10-)Fonksiyonun sonunda yer ayırtılan array belleğe geri verilir.

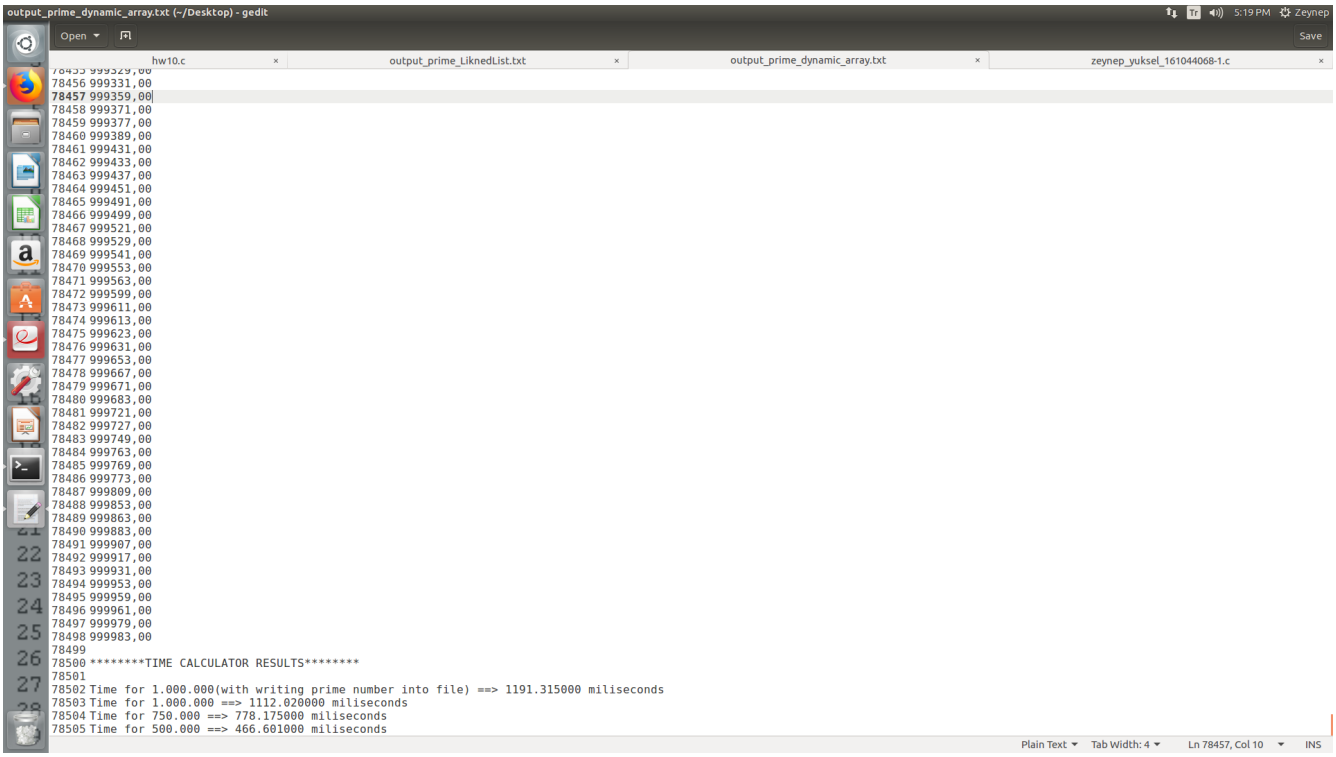
Linked List Adımları :

1-)Data txt dosyası okuma modunda açılır.
2-)Text dosyasından bir tane long int alınır ve while döngüsüne girilir.While döngüsü parametre olarak girilen finalNum sayısı kadar döner.Mesela finalNum 500000 ise data.txt dosyasından 500000 sayısına kadar long int alınır.
3-)Her alınan sayının asal olup olmadıgı kontrol edilir.
4-)Yeni eklencek olan node un datası dosyadan yeni alınan asal sayıdır ve next ide son node olduğu için NULL olur.
5-)Eğer başta linked list boş ise dosyadan alınan sayı ile ilk node oluşturulur.
6-)Eğer eklenecek sayinin önünde başka node lar var ise son node u ifade eden lastNode değişkeni artık yeni eklenecek olan sayının node unu gösterir.
7-)Dosyanın sonundaki EOF un alınmaması için eğer dosyadan alınan bir satırda 3 tane eleman yoksa (yani %ld %c %d) number sayısı 10 artırılarak finalNum dan daha büyük olur ve döngüden çıkar.
8-)Digit Number 1 ise linked listin elemanları bastırılır,0 ise bastırılmaz.
9-)Fonksiyonun sonunda ise node lar free ile belleğe geri verilir.

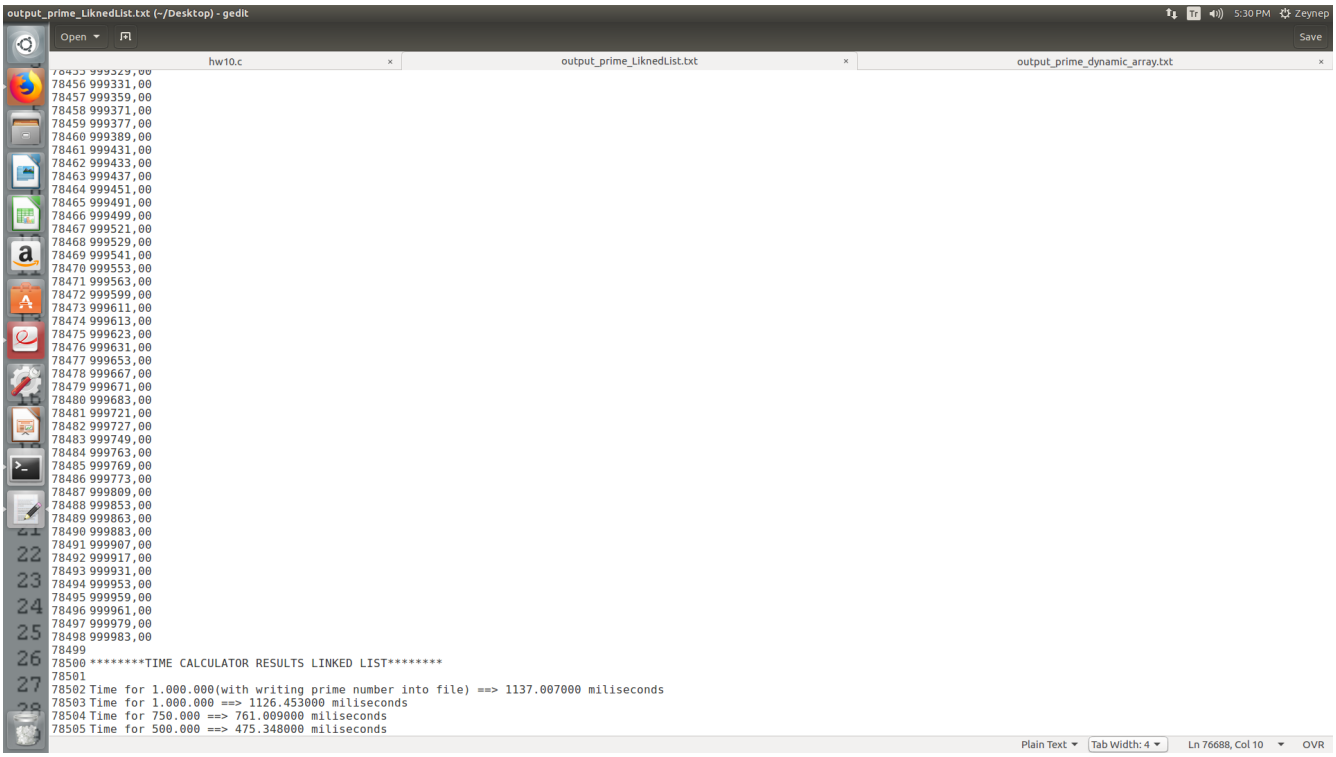
Main Adımları :

1-)Clock fonksiyonu ile zaman hesaplanır : fonksiyona gitmeden ve gittikten sonraki zamanlar birbirinden çıkarılır ve time.h kütüphanesinin sabiti olan CLOCKS_PER_SEC a bölünür.clock fonksiyonu zamanı saniye cinsinden return ettiği için çıkan zaman sonucu 1000 ile çarpılarak milisaniye cinsine dönüştürülür.
2-)Fonksiyonların çalıştıkları zamanlar output_prime_dynamic_array.txt ve output_prime_LiknedList.txt dosyalarına yazılır.Ayreten bu sonuçlar mainde de basılır.
3-)Yazdırılma işlemleri bittikten sonra dosyalar kapanır.

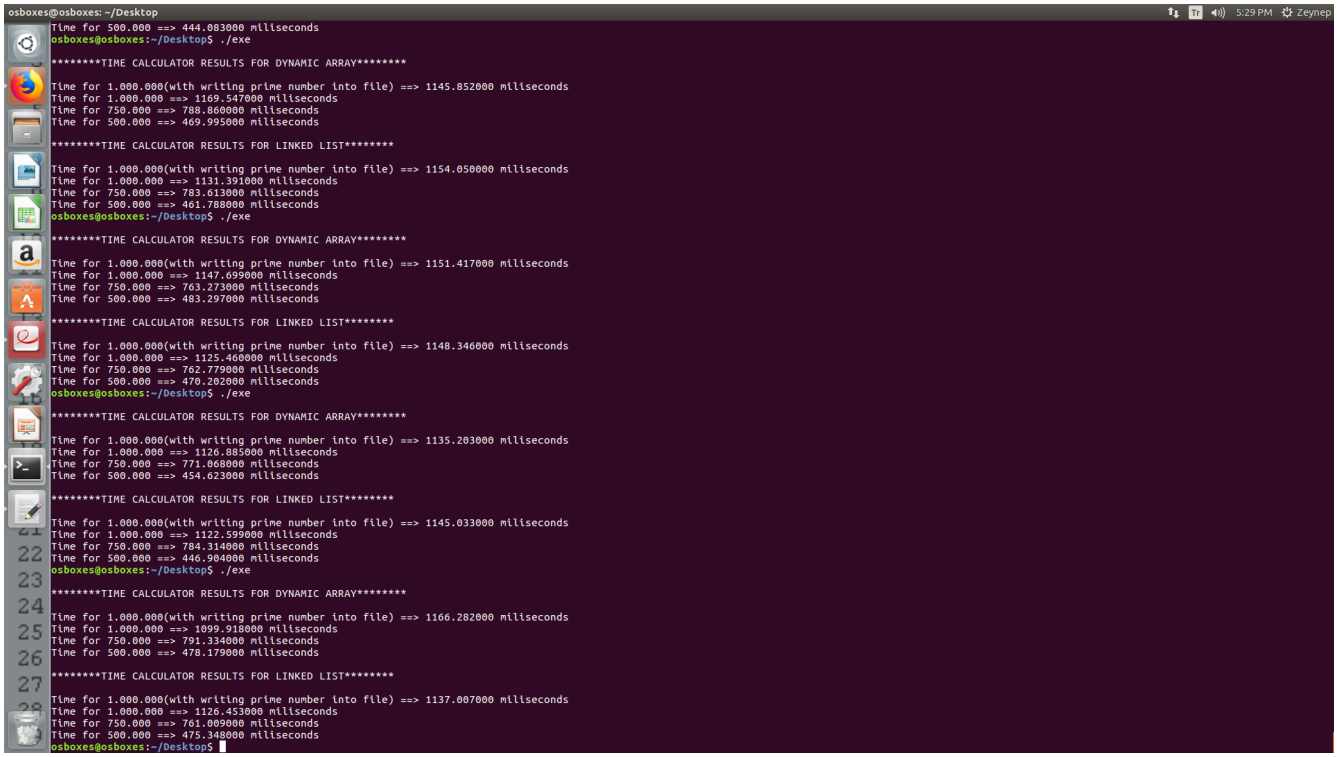
DYNAMIC ARRAY DOSYA ÇIKTISI(output_prime_dynamic_array.txt)



LINKED LIST DOSYA ÇIKTISI(output_prime_LinkedList.txt)



TERMİNAL ÇIKTISI



NOT

1.000.000 ilk çalıştırıldığında asal sayıları dosyaya da yazdırılacağı için daha yavaştır.