**YZM 2124**

Veri Yapıları

2021-2022 Bahar Dönemi

**ÖDEV 1 RAPORU**

Teslim Tarihi: 06.04.2022

Öğrenci Adı: Mustafa Alp YANIKOĞLU

Öğrenci Numarası: 202803035

1. Giriş

Ödevin amacı; Array’de bulunan elemanları güncelleme işlemleri yapıyoruz ve bu güncelleme işlemlerini Bağıntılı Liste özelliklerini kullanarak da yapıyoruz ve bu 2 farklı yolla programın çalışma hızını test edebiliyoruz.

ListNode.h ‘da kullanılan veriler;

int deger : Düğümdeki değeri verir.

Node \*sonraki : Düğümden sonraki elemanın adresini tutar.

LinkedList.h ‘da kullanılan veriler;

Node \*bas : Listenin ilk düğümünü belirtir.

void olustur() : Boş bir liste oluşturur.

Void ekle(int, int) : Birinci parametre ekleyeceğimiz değeri, ikinci parametre ekleyeceğimiz indisi belirler ve listeye bu elemanı ekler.

void guncelle(int ,int) : Birinci parametre ekleyeceğimiz değeri, ikinci parametre güncelleme yapacağımız indisi belirler ve listeyi güncelleriz.

void sil(int) : Gönderdiğimiz parametre silmek istediğimiz indisi gönderir ve listede bulunan o indis silinir.

void yazdir() : Listede bulunan elemanları yazdırır.

Void bosalt() : Listede bulunan tüm elemanları sileriz.

İnt sayac : Listenin kaç elemanlı olduğunu tutar.

Main.cpp ‘de kullanılan veriler;

Liste \*l : İşlemleri yapacağımız liste.

void linkedListProporty() : Linked list ile kullandığımız özellikleri çalıştırır.

void arrayListProporty() : Array ile kullandığımız özellikleri çalıştırır.

ArrayList.h ‘da kullanılan veriler;

int \*bas : Array’in adresini tutar.

void olustur() : Boş bir array oluşturur.

void ekle(int, int) : Birinci parametre ekleyeceğimiz değeri, ikinci parametre ekleyeceğimiz indisi belirler ve array’e bu elemanı ekler.

void guncelle(int ,int) : Birinci parametre ekleyeceğimiz değeri, ikinci parametre güncelleme yapacağımız indisi belirler ve array'i güncelleriz.

void sil(int) : Gönderdiğimiz parametre silmek istediğimiz indisi gönderir ve array’de bulunan o indis silinir.

void yazdir() : Array’de bulunan elemanları yazdırır.

void bosalt() : Array’de bulunan tüm elemanları sileriz.

int sayac : Array’in kaç elemanlı olduğunu tutar.

int tut: Hangi elemanı güncellediğimizi tutar.

2.Geliştirme ve Çalıştırma Ortamları

Geliştirmek ,test etmek ve çalıştırmak için Visual Studio Code editörünü kullandım.

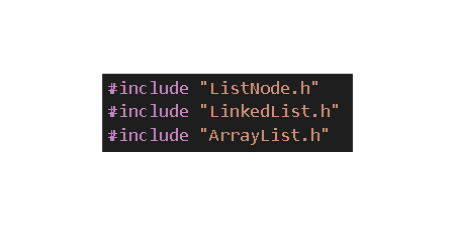
3.Veri Yapıları ve Değişkenler

Bağıntılı liste ve array kullandım. Bunları kullanmamızdaki amaç bağıntılı liste ile dizide yapılabilen işlemleri daha hızlı bir şekilde gerçekleştirebileceğimizi kanıtladık.

Bunların dışında bir de int veri tipi kullandık. Bunu kullanmamızdaki amaçta bazı verileri güncellemeden önce saklamaktı.

4.Program Akışı

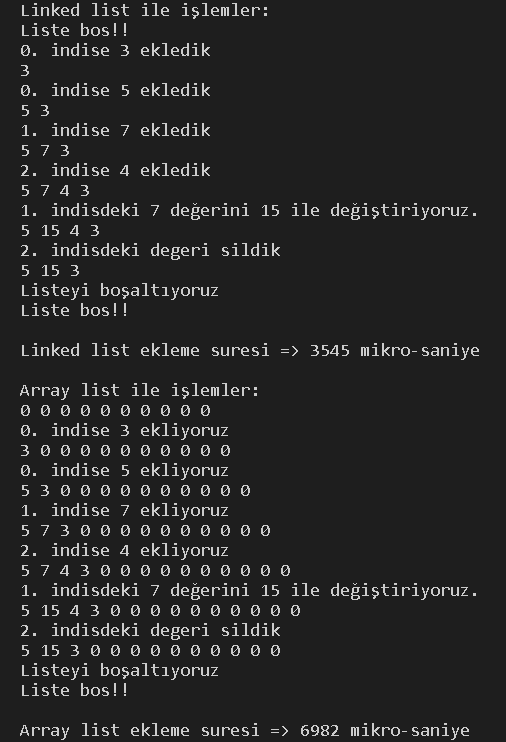
Bu projede 3 farklı kütüphane oluşturduk bunlar:



LinkedList.h’da bağıntılı liste özelliklerini kullanarak ekle, sil gibi işlemleri yaptık. Bu işlemleri yaparken ListNode.h’ da bulunan “int deger” ve “Node \*sonraki” özelliklerini kullandık bunlar sırasıyla istediğimiz değeri ekler ve istediğimiz düğümden sonraki düğümün adresini verir. Bu 2 kütüphanedeki gerekli verileri kullanarak bir listede bulunan indislerin adreslerini düğümler aracılığı ile birbirine bağlayıp listedeki istediğimiz elemanı güncelledik.

ArrayList.h’da ise ilk yaptığımız işlem ‘int \*bas’ pointer ile oluşturduğumuz array’in adresini tuttuk ve bu pointer ile array içinde for döngüsü ile gezip istediğimiz indisteki elemanı güncelledik.

Main.cpp ‘de ise yukarıda anlattığımız iki farklı yoldaki fonksiyonları kullanarak bir listedeki elemanları güncelledik ve bu güncellemeleri “SureHesaplama()” fonksiyonu ile mikro-saniye türünden işlemlerin ne kadar sürdüğünü hesapladık.



5.Sonuç

Bir listede veya dizi de işaretçiler(pointer) sayesinde istediğimiz elemanlara erişebildik istediğimiz elemanları güncelleyebildik ve bağıntılı liste özellikleri ile düğüm(node) oluşturduğumuzda aslında kendinden bir sonraki adresi de tutabildiği için dizilerden daha hızlı liste elemanları güncelleyebildiğimizi gördüm.