**YZM 2124**

Veri Yapıları

2021-2022 Bahar Dönemi

**ÖDEV # RAPORU**

Öğrenci Adı: Hakan Bayazıt Habeş

**1.Giriş**

Ödevin amacı, Arraylist ve linkedlist’te yani iki farklı liste veri yapısında aynı uzunlukta veriler ile aynı indislere veri ekleme, güncelleme, silme, oluşturma, yazdırma ve boşaltma işlemlerini gerçekleyerek bu iki liste veri yapısı arasındaki belleğe yazma ve okuma zamanlarını karşılaştırmaktır.

struct Node {

int deger; 🡪Node’un değerini Tutar

Node\* sonraki; 🡪Node’un Adresini tutar.

};

struct LinkedList {

Node\* bas; 🡪Listenin Başlangıç adresini tutar.

void olustur(); 🡪LinkedList’i oluşturur.

void ekle(int, int); 🡪Listeye ekleme işlemini gerçekleştirir.

void guncelle(int, int); 🡪Listeye güncelleme işlemini gerçekleştirir.

void sil(int); 🡪Listeye silme işlemi Gerçekleştirir.

void yazdir(); 🡪Listeye yazdırma işlemini gerçekleştirir.

void bosalt(); 🡪Listeyi bpşaltma işlemini gerçekleştirir.

int sayac; 🡪Eleman sayısını tutar.

};

struct ArrayList {

int\* bas; 🡪Listenin Başlangıç adresini tutar.

void olustur(); 🡪ArrayList’i oluşturur.

void ekle(int, int); 🡪 Listeye ekleme işlemini gerçekleştirir.

void guncelle(int, int); 🡪Listeye güncelleme işlemini gerçekleştirir.

void sil(int); 🡪Listeye silme işlemi Gerçekleştirir.

void yazdir(); 🡪Listeye yazdırma işlemini gerçekleştirir.

void bosalt(); 🡪Listeyi bpşaltma işlemini gerçekleştirir.

double sayac; };

**2. Geliştirme ve Çalıştırma Ortamları**

Ödev gerçekleştirilirken Windows makine üzerinde ve Visual studio üzerinde oluştur, ekle, guncelle, sil, yazdir, boşalt metotları kullanılarak geliştirilmiştir.

**3. Veri Yapıları ve Değişkenler**

Ödevin gerçekleştirilmesi aşamasında ArrayList ve LinkedList kullanılmıştır. Bu veri yapılarının kullanım amaçları LinkedList’te kullanılan ekleme işleminin ya da silme, güncelleme, yazdırma, boşalt metotlarının kullanırken Node tipindeki değişkenlerin adresler üzerinden nasıl bağlantı sağladıklarını gözlemlemektir. ArrayList’te ekleme işleminin ya da silme, güncelleme, yazdırma, boşalt metotlarının kullanırken int tipindeki değişken ile liste içine nasıl bağlantı kurduğunu gözlemlemektir.

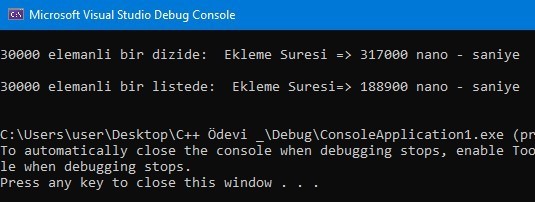
**4. Program Akışı**

1 🡪 Program dosyayı açar ve yazdırır.

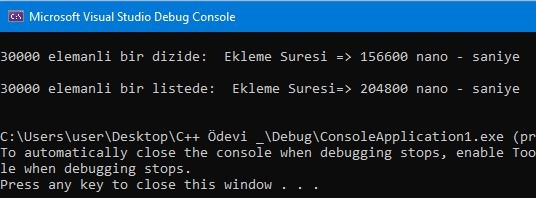
2 🡪Üzerinde LinkedList ve ArrayList veri yapısını kullanarak veriyi saklar.

3 🡪Program ekle() algoritmalarını kullanarak operasyon yapar.

4 🡪Şu çıktıları üretir:



Yukarıdaki çıktıda LinkedList’in ArrayList’ten daha hızlı çalıştığını görmüş olduk.



Burada ise ArrayList’in ekleme süresinin LinkedList’ten hızlı olduğunu görüyoruz. Bunun sebebi üzerinde çalışılan bilgisayarın okuma yazma işleminin hızlı olmasından kaynaklanır. Bunu aşmak için eleman sayısı arttırılabilir.