# VERİ YAPILARI ÖDEV RAPORU

**Ödev sonucu öğrendiklerim:**

1)pointer kullanımı ve fonksiyonlardan pointer kullanarak herhangi bir fonksiyondan diziyi geri döndürmek

2) liste yapımı aşamaları, bu listelerin kullanımı ve listelerin birbirine bağlama

3)veriyi heap bellekte depolama

4)heap ve stack bellek arasındaki farklar ve bunların kullanım yerleri

5)HPP dosyalarının kullanımını ve proje dosyalarını birbirine bağlamayı

6)çalışma hayatımda kullanacağım yazılan programlar için kullanılan dosya düzeni(lib, include, src vs.)

7)Makefile yapımı ve kullanımı

**Ödevde yaptıklarım:** basit bir Node sınıfı oluşturarak başladım ardından circular doubly linked list (CDLL) sınıfını oluşturup constructur ve private kısmını oluşturdum sayılar dosyasının içindeki sayıları çıkarıp programa ekleyen while döngüsünü oluşturdum programım 400 satıra ve sütüna kadar sayılar.txt dosyasını okuyabiliyor. Bu değerlerle CDLL oluşturmak için sınıfın içine son kısma eleman ekleyen instert_last(int) fonksiyonunu tanımladım ardından insert_last(int) fonksiyonunu test için CDLL classının içindeki bütün değerleri okuyan reader() fonksiyonunu tanımladım. Oluşturduğum listeleri birbirine bağlamak için listelerlistesi isimli dizi değişkenini tanımladım ve ardından basit bir while döngüsüyle sayilar.txt dosyasındaki bütün satirlar için bir CDLL class değişkeninin açılıp bunların listelerlistesi değişkenine kaydedilmesini ve ekrana yazdırılmasını sağladım. sayılar.txt kalsöründeki en uzun ve en kısa satırları bulması için 2 fonksiyon(enkisasatiribulma(int* , int) enuzunsatiribulma(int*, int)) tanımladım bu fonksiyonların içine parametre olarak dizi koyamadığım için içine pointer yerleştirdim ve pointer yoluyla diziyi dolaştım ardından çaprazlama kısmına gelince başa sayı ekleyen inster_first(int) baştan sayı silen remove_first() ve sondan sayı silen remove_last() fonksiyonlarını oluşturdum. CDLL sınıfından aldığım değerleri kaydedip bu fonksiyonları doğru bir sırada kullanarak çaprazlama işlemini bitirdim.

**Zorlandığım kısımlar:** sayıları.txt dosyası içerisinden sayıları herhangi bir hataya yer vermeden almak zorlayıcı oldu araya tek boşluk yerine 2 boşluk veya harf konulması durumunda okumanın hata vermesini istemediğimden dolayı kurduğum fonksiyonu düşünmek çok zaman aldı

Aynı zamanda head'in ortada olması ve derste işlediğimiz bütün bağlı listelerde head in başta olması nedeniyle derste yazdığımız fonksiyonlardan daha farklı algoritmalar düşünmek durumdaydım bu da zorlayıcı oldu

Normalde visual studio ile bütün projeyi tek bir sayfada yazdığım için bu yeni kodların ayrı yerlere yazıldığı, HPP ve birden fazla CPP dosyalarının bulunduğu sisteme alışmam uzun sürdü.