



# BSM 207 VERİ YAPILARI – ÖDEV RAPORU

İki basamaklı sayı listesini onlar ve birler  
basamaklarını işleyerek indekslerine göre  
ortalamalarını toplayan program

Osman Emre Turhan ( G211210012 )  
Mahmud Emre Başpehlivan ( G211210020 )  
Osman.turhan@ogr.sakarya.edu.tr ve emre.baspehlivan@ogr.sakarya.edu.tr

## 1. ÖDEV GEREKSİNİMLERİNİN ANALİZİ

Öncelikle ödev dosyasını okuyarak aşağıdaki gereksinimleri çıkardık:

- “Sayılar.txt” dosyası programın main fonksiyonu parametresi ile okutulacak.
- Her satıra ait onlar basamaklarına ait rakamlar tek yönlü bağlı listeye aktarılacak
- Her satıra ait onlar basamaklarına ait rakamlar tek yönlü bağlı listeye aktarılacak
- KonumA değişkeni üst kısma ait satır indeksinin seçilmesini sağlayacak
- KonumB değişkeni alt kısma ait satır indeksinin seçilmesini sağlayacak
- Seçilen indekslerdeki bağlı listeler yer değiştirecek
- Bağlı listelerin (onlar birler ayrı) aynı indeksli elemanlarının ortalamaları alınıp diğer indislere ait elemanların ortalamalarıyla toplanacak
- Sonuç ekrana Ust ve Alt olarak yazdırılacak.

## 2. ÖDEVİN GERÇEKLEMESİ

Ödevi gerçekleştirmek için Discord platformunda ödevi anlamak için biraz tartıştık. Önce programa Sayılar.txt dosyasını okutma kısmını yaptık. Okutulan bu dosyayı gerçekten okuyup okumadığını ekrana çıktılar vererek test ettik. Sonra örnek bir tek yönlü bağlı liste örneği ekledik programa. Bunun için Dugum ve DugumBaglanti sınıflarını oluşturduk. Ust ve Alt diye ayırdığımızda kafamız çok karıştığı için (üstte onlar mı var birler mi var sürekli karışıyordu) programda onlar ve birler olarak değişkenleri kullandık. En son yazdırma kısmında Ust ve Alt olarak ekrana yazdırdık. DugumBaglanti sınıfının içerisinde aslında onlar ve birler için iki tane Dugum sınıfından nesne oluşturduk. DugumBaglanti sınıfının içerisinde bellek sızıntısı olmaması için gerekli yıkıcı fonksiyon tanımlamalarını yaptık. Programın içerisinde bağlı listeye ekleme yapılmasını sağlayan, seçilen bağlı listeleri yer değiştiren ve bağlı listenin seçilmesini sağlayan 3 tane fonksiyon bulunmaktadır. Bu fonksiyonlar sayesinde ödev istenilen değişiklikleri yaparak ekrana getirmektedir. Son olarak onlar ve birler olarak ayırdığımız bağlı listelerin içerisindeki aynı indeksli elemanların ortalamalarını alıp diğerlerinin ortalamasıyla topluyoruz (birler onlar ayrı). Sonucu ekrana yazdırıyoruz. Program bir kere çalışacak şekilde ayarlanmıştır. Ödev içeriğinde tekrar tekrar çalışmasıyla ilgili herhangi bir istek bulunmadığı için böyle yaptık. Bir döngü yapısıyla tekrar tekrar (cikis değeri verilene kadar) çalıştırılabilir.

### MAKEFILE İLE İLGİLİ ÇOK ÖNEMLİ!!!

Makefile dosyası derleme çalıştırma işlemini yapmaktadır. Ancak bin klasöründeki Program.exe dosyası Sayılar.txt parametresine ihtiyaç duymaktadır. Bizim programı test ettiğimiz Sayılar.txt dosyası doc klasörü içerisinde olduğu için Makefile buna göre hazırlanmıştır. Sizin programı çalıştıracağınız Sayılar.txt dosyasını eğer doc klasörüne atarsanız (örnek dosyayı değiştirerek) program çalışacaktır. Ancak bu klasöre koyulmazsa program çalışır ancak sizin denediğiniz Sayılar.txt dosyası yerine örnek dosyayı okur. Ödevde Makefile’ı size en uygun şekilde nasıl ayarlayabileceğimle ilgili bir açıklama bulamadığım için bu açıklamayı yapıyorum.

### 3. ÖDEVİ YAPARKEN ZORLANDIĞIMIZ YERLER

Ödevi yaparken en çok zorlandığımız 2 yer vardı:

- Bağlı listeleri satır özelinde ayırmak
- Bağlı listelerin indekslerinin ortalamalarını alıp diğerleriyle toplamak

Bağlı listeleri liste özelinde ayırmak için 2 farklı yöntem denedik birincisi iki ayrı sınıf yazmak (onlar ve birler için) ikincisi tek bir sınıf ile ikisini de yapmak en son olarak tek sınıfta yaptık. Aynı şekilde yapıyı bu şekilde oturtunca ikinci sorunu da çözmek kolaylaştı.

### 4. SINIF BAŞLIKLARI VE PROGRAM ÇIKTISI

```
#ifndef DUGUMBAGLANTI_HPP
#define DUGUMBAGLANTI_HPP

#include "Dugum.hpp"
#include <iostream>

class DugumBaglanti {
public:
    Dugum* onlarList;
    Dugum* birlerList;
    DugumBaglanti* next;

    DugumBaglanti();
    ~DugumBaglanti();
};

void BagliListeyeEkle(Dugum*& head, int data);
void IndexIleBagliListeDegistir(DugumBaglanti* head, int onlarIndex, int birlerIndex);
DugumBaglanti* IndexIleDugumBaglantiAl(DugumBaglanti* head, int index);

#endif // DUGUMBAGLANTI_HPP
```

```
#ifndef DUGUM_HPP
#define DUGUM_HPP

class Dugum {
public:
    int data;
    Dugum* next;

    Dugum(int d);
};

#endif // DUGUM_HPP
```

```
Konum A: 5
Konum B: 10

Ust: 74.7341
Alt: 67.4484
```