

# 1. Ödev

[Tüm Şubeler İçindir, Ara Sınav Olarak Değerlendirilecektir]

Teslim Tarihi

16 Ağustos 2020 (Pazar günü saat 23:59'a kadar.)

## Ödev İçeriği

Yazacağınız program C++ konsol uygulaması olup çalıştırıldığında Sayilar.txt isimli dosyayı okuyacaktır. Bu dosya içerisinde her satırda 0-255 arası pozitif sayılar bulunacak ve sayıları birer boşluk ayıracaktır. Dosyadaki her satır için bir adet **iki yönlü dairesel bağlı liste** nesnesi oluşturulacaktır. Sırayla sayılar okunup aşağıdaki algoritmaya göre listelere eklenecektir.

- En büyük OBEB (Ortak Bölenlerin En Büyüğü) hiç sayı okunmadığı için başlangıçta sıfır alınacak.
- Örneğin 82 sayısı geldi ilk sayı olduğu için direk listeye eklenir. En büyük obeb sıfır kabul edilir.  
| 82 | En Büyük OBEB=0
- 75 sayısı geldi
- 75 ile 82'in OBEB'i 1 olduğu için şimdiye kadarki en büyük obeb 0 olduğundan bu düğümde kalıyoruz hangi sayı büyükse mod alıyoruz  $82 \text{MOD} 75 = 7$  0 dışında bir sayı gelirse sol tarafa gelen sayı kadar ilerliyoruz, 0 gelirse bir sağına sayıyı ekliyoruz. 7 geldiği için 7 kere sol yapılacak fakat liste başına gelinirse eleman oraya eklenir.  
| 75 | <-> | 82 | En Büyük OBEB=1 (Şimdiye kadarki)
- 65 geldi
- 75 65 obeb=5  $5 > 1$  olduğu için bu düğümde kalıyoruz.  $75 \text{MOD} 65 = 10$  10 adım sola ilerliyoruz. Liste başına gelindiği için liste başına yeni gelen sayıyı ekliyoruz.  
| 65 | <-> | 75 | <-> | 82 | En Büyük OBEB=5
- 83 geldi
- 65 83 obeb=1  $1 < 5$  olduğu için devam ediyoruz
- 75 83 obeb=1  $1 < 5$  olduğu için devam ediyoruz
- 82 83 obeb=1  $1 < 5$  olduğu için devam ediyoruz fakat liste sonuna gelindiği için bu düğümde kalıyoruz.
- $83 \text{MOD} 82 = 1$  sol tarafa 1 düğüm ilerle  
| 65 | <-> | 75 | <-> | 83 | <-> | 82 | En Büyük OBEB=5
- 65 geldi
- 65 65 obeb=65  $65 > 5$  olduğu için bu düğümde kalıyoruz.
- $65 \text{MOD} 65 = 0$  düğümün sağ tarafına ekliyoruz.  
| 65 | <-> | 65 | <-> | 75 | <-> | 83 | <-> | 82 | En Büyük OBEB=65
- 121 geldi
- 65 121 obeb=1  $1 < 65$  olduğu için devam ediyoruz.
- 65 121 obeb=1  $1 < 65$  olduğu için devam ediyoruz.
- 75 121 obeb=1  $1 < 65$  olduğu için devam ediyoruz.
- 83 121 obeb=1  $1 < 65$  olduğu için devam ediyoruz.
- 82 121 obeb=1  $1 < 65$  olduğu için devam ediyoruz. Fakat liste sonuna gelindiği için bu düğümde kalıyoruz.
- $121 \text{MOD} 82 = 39$  39 adım sola ilerliyoruz. Liste başına gelindiği için orada kalıyoruz.

| 121 | <-> | 65 | <-> | 65 | <-> | 75 | <-> | 83 | <-> | 82 | En Büyük OBEB=65

- 65 geldi
- 65 121 obeb=1 1<65 olduğu için devam ediyoruz.
- 65 65 obeb=65 65=65 olduğu için kalıyoruz.
- 65MOD65=0 düğümün sağ tarafına ekliyoruz.

| 121 | <-> | 65 | <-> | 65 | <-> | 65 | <-> | 75 | <-> | 83 | <-> | 82 | En Büyük OBEB=65

Dosyadaki ilk satırın burada tamamlandığını düşünürsek, listede oluşan sayıların ASCII karşılıkları karakterler ekrana yazılır. Yukarıdaki örnekte sayılara karşılık gelen karakterler ekrana yazıldığında

Sifre: yAAAKSR

Dosyadaki her bir satır ayrı bir şifreyi temsil eder. Program çalıştırıldığında yukarıdaki algoritmaya göre hesaplama yapacak ve bütün şifreleri ekrana yazıp program sonlanacaktır. Hesaplamalar ekrana yazılmayacaktır. **Dosyadaki satır sayısı ve her satırdaki sayı adedi değişken sayıda olabilir.** Büyük boyutlu (örneğin 500 satırlı ve her satırında en az 250 sayı bulunan) dosyalarda denenmelidir.

Ödevde hazır veri yapısı kullanılamaz, her öğrenci ödevde uygun veri yapısını tasarlamalıdır. Veri yapıları heap bellek bölgesinde oluşmalıdır. Program süresince **çöp oluşmamalıdır.** **Generic (Şablon) yapı kesinlikle kullanılmamalıdır.**

**Başlık ve kaynak dosyaları her sınıf için mutlaka ayrı olmalı ve başlık dosyasında metot gövdesi bulunmamalıdır.**

**Önemli Not:** Raporunuz detaylı olmalı ve kendi cümleleriniz olmalıdır. Kopya ödevler sıfır olarak değerlendirilecektir. SABİS şifreniz sizin sorumluluğunuz altındadır eğer arkadaşınız sizden habersiz ödevinizi alırsa bundan sizde sorumlu tutulur ve sıfır alırsınız.

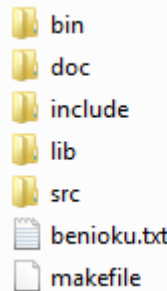
## Teslim Formatı

Klasör Hiyerarşisi

- src Klasörü (Kaynak dosyalarınızın bulunduğu klasör)
- **lib Klasörü** (.o dosyalarınızın bulunduğu klasör)
- include Klasörü (Başlık dosyalarınızın bulunduğu klasör)
- doc Klasörü (Raporunuzun ve eğer varsa diğer dokümanların bulunduğu klasör.)
- bin Klasörü (Çalıştırılabilir dosyalarınızın (.exe) bulunduğu klasör.)
- benioku.txt (Program için ayrıca belirtmek istedikleriniz. Önerileriniz. Boş kalabilir.)
- **makefile**

**makefile dosyası olmayan ödev derlenmemiş olarak kabul edilmektedir. (-50 puan)**

**Rapor Yoksa veya yetersiz ise (-15 Puan)**



Rapor pdf formatında olmalıdır. Raporunuzda kısaca sizden istenilen, öğrendikleriniz, ödevde yaptıklarınız, eksik bıraktığınız yerler, zorlandığınız kısımlar anlatılabilir. **Ödev raporunda yazı boyutu 12 puntodan büyük olamaz ve en az 2 sayfa en çok 4 sayfa olabilir.**

Klasörlerinizi bir ana klasör içine koyarak .rar veya zip'leyip **SABİS** üzerinden gönderiniz. Yukarıda belirtilen teslim tarihinden sonra gönderilen ödev **kesinlikle kabul edilmeyecektir.**

**Önemli: SABİS üzerinden bir ödev için dosya gönderme hakkınız 3 seferdir. Hoca ekranında en son yüklediğiniz dosya görünür.**

Mail ile gönderilen ödevler **kesinlikle kabul edilmeyecektir.**

Yazmış olduğunuz bütün kaynak kodların en başında aşağıdaki bilgiler bulunmalıdır. Bilgileri kendinize göre güncelleyiniz.

```
/**
 * @file          Dosya adı
 * @description   Programınızın açıklaması ne yaptığını dair.
 * @course        Dersi aldığınız eğitim türü ve grup
 * @assignment    Kaçınıcı ödev olduğu
 * @date          Kodu oluşturduğunuz Tarih
 * @author        Öğrencinin Adı Soyadı
 */
```

**Önemli: Yazacağınız kodlar MinGW'de kesinlikle derlenmelidir. Derlenmeyen kod itibar görmez ve çok düşük puan üzerinden değerlendirilir. Sadece C++ kodları kullanılmalıdır. Ödev bireyseldir.**

**KOPYA ÖDEV SIFIR OLARAK DEĞERLENDİRİLMEKTEDİR\***

**\* Şubeler farklı dahi olsa kopya ödev gönderenler sıfır alacaklardır.**