

1. Ödev [Tüm Şubeler İçin]

Teslim Tarihi

24 Kasım 2022 (Perşembe günü saat 23:59'a kadar.)

Ödev İçeriği

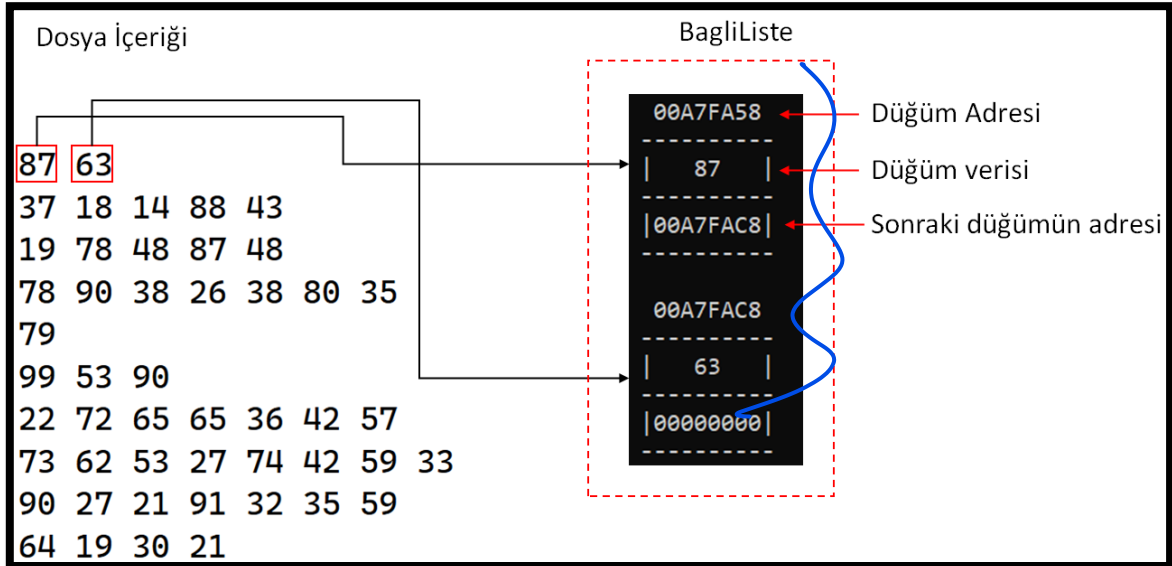
MinGW C++ ortamında program yazılacaktır.

Bu programda Sizlerden bir çift yönlü bağlı liste uygulaması yapmanız istenmektedir.

Geliştireceğiniz uygulamanın içeriği ve özellikleri aşağıda belirtilmiştir.

Aşağıdaki şekilde her satırda birden fazla tam sayı barındıran bir dosya içeriği verilmiştir. Satır sayısı ve her satırdaki sayı adeti dosyadan dosyaya değişmektedir.

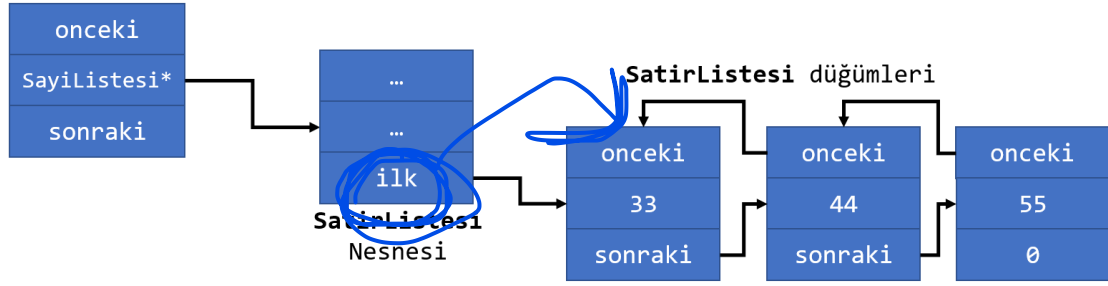
- Bu dosyada her bir satırdaki sayılar bir bağlı listenin düğümlerinde tutulacaktır.
- Bu listelerin sınıfı **SatirListesi** ismi ile tanımlanmalıdır.



Bir satırdaki sayılar için oluşturulan **SatirListesi**'nin Gösterimi

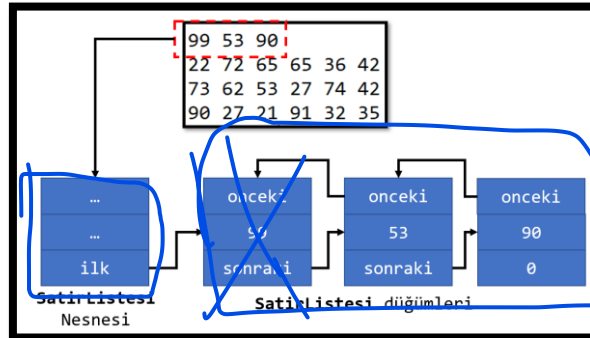
- Oluşturulacak **SatirListesi** nesnelerini yönetmek içinde başka bir bağlı liste kullanılacaktır.
- Bu bağlı listenin sınıf ismi de **YoneticiliListesi** olarak adlandırılmalıdır.
- YoneticiliListesi**'nin her bir düğümü bir **SatirListesi** nesnesinin adresini tutacaktır.

Yoneticilistesi için Düğüm



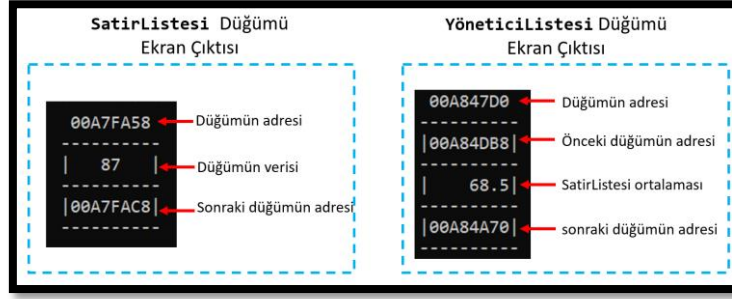
Yoneticilistesi ile Sayilistesi arasındaki ilişki

- Uygulamanız çalıştırıldığında proje ana klasörü içerisinde bulunan "[veriler.txt](#)" dosyası okunacaktır.
DİKKAT: Dosyayı indirerek kullanın. Tarayıcı üzerinden seçip yapıştırmayın sorun yaşayabilirsiniz.
<https://drive.google.com/file/d/1Q7dVUqI5XIViPJcbi-OufQir0bdpWq9S>
- Tüm **SatirListesi** nesnelerini barındıracak bir adet **Yoneticilistesi** nesnesi heap alanında oluşturulacaktır.
- Her bir satır için öncelikle bir adet **SatirListesi** nesnesi heap üzerinde oluşturulacak ve satırdaki sayılar bu listeye eklenecektir.



Bir satır sayı kullanılarak oluşturulan SatirListesi

- Satır okuma bittikten sonra oluşturulan **SatirListesi** nesnesi **Yoneticilistesi**'ne eklenecektir.
- Yoneticilistesi**'ne ekleme işlemi yapılırken eklenecek **SatirListesi** nesnesinin sahip olduğu sayıların aritmetik ortalamasına göre ekleme yapılacaktır.
- Aşağıdaki şekilde örnek bir **Yoneticilistesi** gösterilmiştir. Her bir düğümün gösterdiği **SatirListesi** içerisindeki sayıların ortalaması küçükten büyüğe doğru gitmektedir.



SatirListesi ve YöneticiListesi düğümlerinin ekran görüntüsünün

- Kullanıcı 'z' tuşuna bastığında seçili düğüm bir öncekine giderken 'c' tuşuna bastığında bir sonraki düğüm seçilmelidir. Örnek Animasyon Linki: <https://drive.google.com/file/d/1LEIS4u00JUI5IJJEHTpy1ONAx0yv5b5b>
- Program çalıştırıldığında ekranda **YöneticiListesi** düğümlerinden 8 düğüm gösterilecektir (gösterilecek düğüm sayısı 8'den az olmadığı durumlar hariç).
- Kullanıcı 'a' ve 'd' tuşları ile tüm listeyi 8'er olarak dolaşabilecektir.
- Örneğin toplamda 18 düğüm varsa başlangıçta 0-7 düğümleri gösterilecektir. Kullanıcı 'd' tuşuna bastığında 8-15 düğümleri ekrana çıkarken bir sonraki 'd' tuşunda 16-17 düğümleri ekrana çıkacaktır. Aşağıdaki linkten örnek animasyonu izleyebilirsiniz. https://drive.google.com/file/d/19dfsV0vtNvI2PWgoQ_AR49b4FPiMhguO
- Kullanıcı 'p' tuşuna bastığında seçili yolan düğüm **YöneticiListesinden** çıkartılacaktır. <https://drive.google.com/file/d/1LM3P7ths5xjC7bAIFlOuzE3K31XcvMIU>
- Kullanıcı 'k' tuşuna bastığında seçili olan **düğüme** ait **SatirListesi** içerisinden rastgele bir düğüm silinmek için seçilecektir.
- Tekrar 'k' tuşuna basıldığında bu düğüm silinecektir.
- Silme işlemi gerçekleştikten sonra ilgili **SatirListesi** nesnesinin ortalama değeri değişeceğinden **YöneticiListesi** içerisindeki konumu da buna göre güncellenmelidir. Örnek Animasyon linki: <https://drive.google.com/file/d/1LRLEfE72QVCCy63fiygzjoKnwZRpAZXK>
- Düğümü silinen **SatirListesi** nesnesin hiçbir düğümü kalmadıysa bağlı olduğu düğümü de **YöneticiListesinden** çıkartılmalıdır. Örnek Animasyon Linki : <https://drive.google.com/file/d/1NR5Z4Qm5bpqyT1MX1ROyCF7tCQNYQk2>

Programın çalışmasını gösteren videonun linki aşağıdadır.

https://drive.google.com/file/d/1U4uX8F_7jLCR4K6Gw_dQVkvXi46Uvnir

Ekran Çıktılarını elde ederken aşağıdaki kod parçasını incelemek işinize yarayacaktır.

Dikkat :Kopyalayıp yapıştırırsanız problem çıkabilir

```
int index = 1;

while (true)
{
    //Her işlemten sonra ekran temizlenecek. Sadece windows işletim sisteminde çalışır.

    system("cls");

    //Daha sonra güncel görüntü ekrana çıkarılacak
    //setw istenilen sayıda boşluk bırakmak için kullanılabilir.
    cout << setw(index) << " " << "^" << endl;
    cout << setw(index) << " " << "|" << endl;
```

```
//Kullanıcı girişi okunacak.  
char ch;  
cin >> ch;  
//Kullanıcı girişine göre sisteme güncelleme yapılacak  
if (ch == 'a')  
    index--;  
else if (ch == 'd')  
    index++;  
}
```

Çok büyük dosyaları okuyabilecek şekilde program yazılmalıdır. Programda çöp oluşması puan kırılmasına neden olur.

Hiçbir şekilde Şablon (Generic) veri yapısı kullanılamaz. Her sınıfın başlık ve kaynak dosyası ayrı olmalı ve başlık dosyasında metot gövdesi bulunmamalıdır. Hazır veri yapısı kullanılamaz.

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.

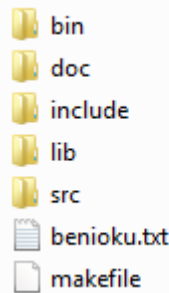
Teslim Formatı

Klasör Hiyerarşisi

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

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

Rapor Yoksa veya yetersiz ise (-15 Puan)



Rapor pdf formatında olmalıdır. Raporu ayrıca çıktı olarak getirmenize gerek yoktur. Raporunuzda kısaca sizden istenilen, öğrendikleriniz, ödevde yaptıklarınız, eksik bıraktığınız yerler, zorlandığınız kısımlar anlatılabilir. **Ödev raporunda yazı boyutu 12 puntodan büyük olamaz ve en az 1 sayfa en çok 2 sayfa olabilir.**

Klasörlerinizi bir ana klasör içine koyarak .rar veya zip'leyip **SABİS** üzerinden gönderiniz. Ana klasörünüzün adı sadece sizin öğrenci numaranız (örn: B111210090) olmalıdır. 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.

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     Kaçıncı ödev olduğu
 * @date           Kodu oluşturduğunuz Tarih
 * @author         Gruptakilerin yazar adları ve mail adresleri
 */
```

Önemli: Yazacağınız kodlar MinGW derleyicisinde 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.**