

2. Ödev [Tüm Şubeler İçin]

Teslim Tarihi

24 Aralık 2023 (Pazar günü saat 23:50'ye kadar.)

Ödev İçeriği

MinGW C++ ortamında derlenen bir program yazılacaktır. Program çalıştığı gibi Veri.txt dosyasından sayıları okuyacaktır. Her satırda değişken sayıda birer boşluk ile ayrılmış pozitif tam sayılar (int) bulunmaktadır. Her satır ayrı bir AVL ağacına yerleştirilecek. AVL ağacında olan bir değer bir daha gelirse AVL ye **eklenmeyecektir**. Her AVL ağacının **sadece yaprakları** postorder okuma ile ayrı bir Yığıtaya yerleştirilecektir. Yığıtaya yerleştirilmeyen sayıların her AVL ağacı için ayrı ayrı düğüm değerlerinin toplamaları bulunacak ve ayrı ayrı aşağıdaki formüle gönderilecektir.

$$\text{int ascii} = \text{AVLToplamDugumDeger} \% (90 - 65 + 1) + 65;$$

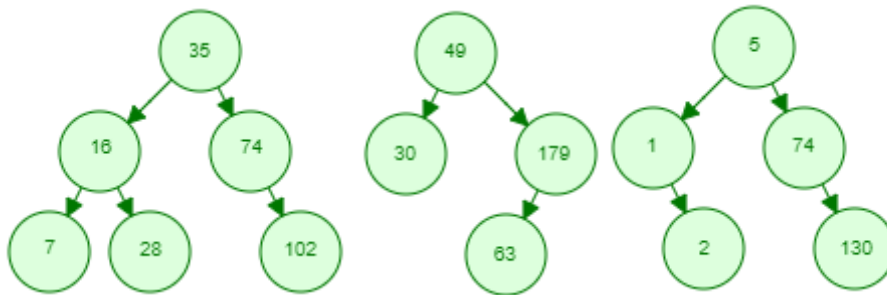
İlk oluşan AVL ağacından son oluşan AVL'ye doğru bu ascii değerlerinin karşılık geldiği karakterler ekrana bitişik yazdırılacaktır. Yığıtlardan çıkmak üzere olan sayılardan ilk olarak **en küçük olanı** çıkarılacak eşitlik durumunda ilk oluşturulan yığıttan veri çıkarılacak daha sonra **en büyük olanı** çıkarılacak eşitlik durumunda ilk oluşturulan yığıttan veri çıkarılacak bu işlem elemanı biten ilk yığıtaya kadar **bir en küçük bir en büyük** olarak devam edecektir. Hangi AVL ağacı ile ilişkili yığıt önce bitmiş ise o AVL ağacı yığıtı ile birlikte yok edilecektir.

AVL ağaçlarının yapraklarının postorder okuma ile yığıtlara yerleştirilme işlemi ve devamındaki işlemler tek bir AVL ağacı kalana kadar tekrar edilecektir. Son kalan AVL ağacının yaprakları hariç yukarıdaki formülle elde edilen karakteri Son Karakter ifadesi ve kaçınıcı AVL olduğu (AVL numaraları 1'den başlar.) ekrana yazdırılacak ve program sonlanacaktır. Referans almak adına aşağıdaki örneği inceleyiniz.

Örnek: Veri.txt içeriği

```
102 74 16 35 7 28
30 49 179 63
5 1 74 130 2
```

Oluşan AVL ağaçları



Yığıtlar

102 28 7	63 30	130 2
----------------	----------	----------

Düğüm Değerleri Toplamları

AVL1: 125 AVL2: 228 AVL3: 80
 Ascii1: 86 Ascii2: 85 Ascii3: 67

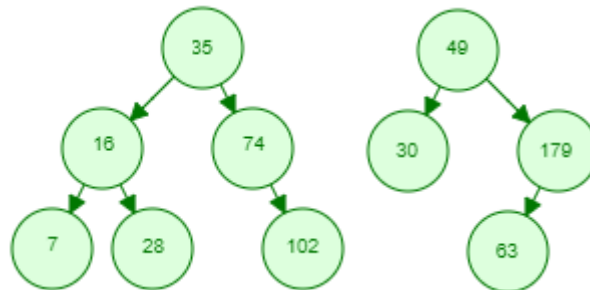
O anki Ekran Çıktısı: **VUC**

Yığıtlardan çıkmak üzere olan sayılardan ilk en küçük olanı çıkarılacak eşitlik durumunda daha önce oluşturulan yığıttan veri çıkarılacak daha sonra en büyük olanı bu işlem en küçük, en büyük şeklinde elemanı biten ilk yığıta kadar devam edecektir.

Yığıtlar

102 28 7	63 30	130 2	
102 28 7	30	130 2	En küçük
102 28 7	30	2	En Büyük
102 28 7	30		En küçük

3 nolu AVL ağacı yok edilecektir.



Yığıtlar ilk durumlarındaki gibi bu tura başlayacaklardır.

Yığıtlar

102 28 7	63 30
----------------	----------

Düğüm Değerler Toplamları

AVL1: 125 AVL2: 228

Ascii1: 86 Ascii2: 85

O anki Ekran Çıktısı: **VU**

Yığıtlar

102 28 7	63 30
----------------	----------

102 28 7	30
----------------	----

En Küçük

28 7	30
---------	----

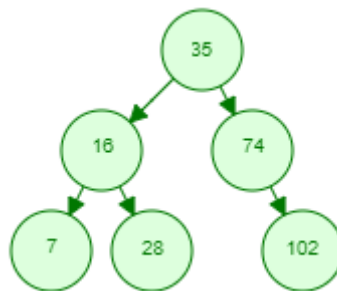
En Büyük

7	30
---	----

En Küçük

7	
---	--

En Büyük



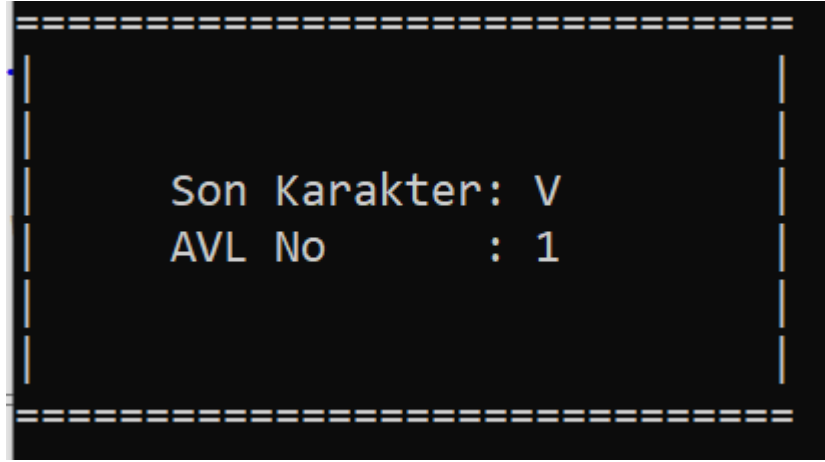
2 nolu AVL ağacı yok edilecektir.

Tek AVL ağacı kaldığı için ekrana

Son Karakter: V

AVL NO: 1

şeklinde yazdırılacaktır.



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

500 satırlık örnek bir dosya SABİS dokümanlara eklenmiştir. Örnek ekran çıktısı aşağıdaki videoda verilmiştir. Programınızın verdiği çıktı birebir aynı olmalıdır.

<https://youtu.be/Af4uBPXxs9c?si=w7CFkvksSgihl2mz>

Bu 500 satırlık örnek dosyayı programınız **maksimum 40 saniyede** çözmelidir. Daha uzun süren çözümlerde kullanılan veri yapıları ve algoritmanın üzerinden geçilmelidir.

Hiçbir şekilde Şablon (Generic) veri yapısı kullanılamaz. Her sınıfın başlık ve kaynak dosyası ayrı olmalı ve başlık dosyasında metod 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.

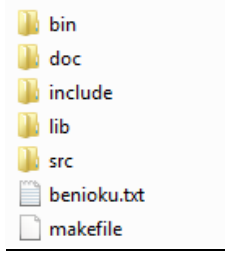
Teslim Formatı

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 yaptığınıza 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 olarak değerlendirilecektir.**