RAPOR

İlk olarak eksik bıraktığım, daha doğrusu yarım bıraktığım Yığıt kısmından bahsedeceğim. Yığıt oluşturmak için "Stack" adında bir class tanımladım ve gerekli tüm fonksiyonları bu class içerisine ekledim. Fonksiyonların çalıştığını ayrıca denedim ve tamamı sorunsuz çalışmakta. "Node" classı içerisinde her düğüme özel bir yığıt olacak şekilde tanımlama yaptım(Node.hpp 27.satır). Daha sonra "ADD.cpp" içerisinde "AddStk" adında bir foksiyon ile (91-109 Satırlar) var olan ağacın düğümlerinde postorder dolanarak her düğümün güncel yüksekliğini, düğüme ilk eklendiğinde olan yüksekliği ile karşılaştırarak, o düğümün yığıtına gerekli harfi eklemeyi hedefledim. Eklemeleri gerçekleştiriyorum ancak istenen sonucu alma konusunda başarısız oldum. **Fonksiyonların çalışmasında bir sorun yok, ancak yığıta yapılacak olan harf eklemeleri ve düzen konusunda bir yordam oluşturamadım.** Bunu takiben pop ederek yazdırmayı integer değerlerde başarılı bir şekilde gerçekleştirdim ancak yine bu yazdırmayı ödevime ekleme konusunda eksik kaldım. Bu iki problem dışında sorunsuz çalışan bir ödeve sahibim.

Çalışma şekline ve dosya içeriklerine geçecek olursak, bir "Node" class'ına sahibim. İçerisine oldukça fazla değişken ekledim. İsim,age ve kilo değişkenlerini, txt dosyasında yer alan 3 farklı bilgiyi tutarak düğüme eklemek için kullandım. Herhangi bir işlem içerisinde yer almıyorlar. Yas değişkenim ise düğüme ekleme yaparken kullandığım yaş parametresini barındırıyor. Bunu düğümün bir özelliği olarak ekleyerek kolayca kontrollerimi sağladım ve gerekli eklemeleri, dengelemeleri kolayca yaptım. Oldheight değişkenim ise Yığıta yapılacak eklemelerde kontrol için, yeni oluşan düğümün ilk yüksekliğini tutuyor. Node class'ım diğer tüm class ları bir nevi birleştirmektedir.

"ADD" class ise düğüm oluşturma, düğümleri dengeleme, yazdırma, yığıt oluşturma gibi geriye kalan tüm fonksiyonları bünyesinde barındırıyor. Burada eksik kalan noktaları raporumun başında anlattım. Derste anlattığınız ve gösterdiğiniz üzere, yığıtın kontrolünün bende olması amacıyla fonksiyonları private alanda tanımladım. Burada beni en çok zorlayan kısımlar düğümler üzerindeki dengeleme kısımları oldu. Kod yazımı bitişinde bir hata almadım ancak ilk olarak konuyu anlamam ve bu anladıklarımı ödeve uygun bir koda dökmem biraz zamanımı aldı. Burada, derste anlattığınız özyineleme ile paralel bir şekilde ilerlemeyi tercih ettim. Özyineleme konusunda oldukça büyük eksiklerim vardı. **Ağaç veri yapısını anlamak ile beraber özyineleme konusunu da bu ödevle beraber büyük ölçüde anlamış, eksiklerimi gidermiş bulunuyorum.** Avl ağacı ile ilgili olan tüm fonksiyonlar sorunsuz çalışmaktadır. Ağaç oluştururken herhangi bir hata veya zorluk ile karşılaşmadım. Yalnızca basit kod yazım hataları. En karmaşık ve uzun olan class'ı öğrenerek ve kolayca tamamladım.

Main fonksiyonunun bulunduğu "calistir.cpp" dosyasında ise ilk olarak txt okuma ve "#" e göre verileri ayırma işlemini gerçekleştirdim. Getline metodu ile txt dosyasını satır satır okuyarak her satırda "strtok" ile isim, yaş, kilo bilgilerini ayırdım. Bu ayırma işlemini if(x) kontrolünü gerçekleştirerek, x==o ise isim yani ilk "#" işaretine kadar olan veri, sonrasında x==1 doğum yılı ve x==2 kilo olarak değişkenlerimin içerisine atadım. Sonrasında "yas" değişkenimin içerisine kişinin doğum yılını integer değere çevirerek yaş hesabı şeklinde

atadım. Daha sonra ise 3 adet değişkenimi ve bunların yanında integer değer olan yaş değişkenimi kontrol amaçlı kullanılması için add fonksiyonuna gönderdim. Burada özyineli olarakbir ekleme gerçekleşiyor. Eğer yeni gönderdiğim yaş değeri kök değerinden küçük ve eşit ise sola büyük ise sağa giderek ekleme devam ediyor. Devamında sizlerin derslerinden ve kaynaklarından yararlanarak dengeleme kısımlarını yaptım. Bu konuda beni en çok zorlayan kısım burası olmuştur. **Konuyu birden fazla kez sizin anlatımınızdan dinledim, ağaca eklemeyi kağıt üzerinde defalarca kez yaptım ve en sonunda sizlerin de kaynaklarından yola çıkarak ekleme ve dengeleme kodlarını tamamladım.** Bu konuyu böylece pekiştirmiş ve öğrenmiş oldum. Main fonksiyonunun devamında ise postorder yazdırma fonksiyonunu ve diğer çöp bırakmama fonksiyonlarını çağırarak programımı sonlandırdım.

Genel olarak AVL ağacı konusunda oldukça büyük ilerleme kaydettiğimi düşünüyorum. Gerek kod yazım sırasında gerekse mantığını anlamak konusunda ödevi yapacak yeterliliğe ulaştım. Ödevimin çalışmayan kısmı için ise son güne kadar çabaladım. O yüzden ödevi size son gün atıyorum. Yeteri kadar vaktimin olmaması Yığıt konusunda eksiğimin olmasına yol açtı. Bu sizin verdiğiniz vakitten değil, benim sağlık sorunlarım ile ilgili olan vakit ayıramama eksikliğidir. Oldukça fazla yöntem denedim. Bunlardan en başarılısı “add” fonksiyonu içerisinde gerekli harf atamalarını yapmak oldu ancak burada “D” harfini yığıta ekleme kısmını yapamadım. Sonrasında şu an da kullanmış olduğum, düğümün köke olan uzaklığındaki değişimlere göre yığıt oluşturma yöntemini denedim ve gördüğünüz, göreceğiniz üzere tam anlamıyla çalışmamaktadır. Tüm çabalarınıza rağmen eksik bir ödev gönderdiğim için affınıza sığınıyorum. İlerleyen zamanda ödevimin eksik kısmını tamamlayacağımın garantisini vermek isterim.

Okulumda (Ondokuz Mayıs Üniversitesi) C dilini kullanmamız ve yaz okulu döneminde c++ görüyor olmamız hem iyi hem kötü yönde sonuçlar doğurdu. Büyük bir yoğunluk ile c++ ve Makefile ile derleme konusunda oldukça bilgi sahibi oldum. Ödevim özelinde ise çeşitli zorluklar yaşadım ancak Metodları ayırırken harici bir problem ile karşılaşmadım.

Bunlara ek olarak sizin verdiğiniz iki örneği de denedim ve sorunsuz çalıştı. Yine kendim, yeni veriler oluşturarak kağıt üzerinde bir yerleştirme yaptım ve programım beni doğruladı. Yine derste göstermiş olduğunuz avl ağacı ekleme uygulamasından de bu durumu teyit ettim.

Makefile dosyası ve ödevim sorunsuz bir şekilde derlenmektedir. AVL ağacına ekleme kısmı sorunsuz çalışmaktadır. Yığıt konusundaki tüm eksikliklerimi belirtmiş bulunmaktayım.

Ödevimi değerlendirmenize arz ederim.