

BSM207 VERİ YAPILARI ÖDEV1 RAPOR

Ödevde İstenen

Bu ödevde, tek yönlü bağlı listeler kullanarak tam sayılar üzerinde çeşitli işlemler yapmamız isteniyor. Yazılan programın MinGW C++ ortamında derlenmesi istenmiştir. Program Sayilar.txt içindeki tam sayıları okuyarak, her bir sayıyı bir bağlı liste olarak saklayacaktır. Bu bağlı listede, her bir basamak bir düğüm olarak temsil edilecek ve bu düğümler 'Basamak' adlı bir sınıf içinde tanımlanacaktır. Aynı şekilde, tam sayıları içeren bağlı liste 'Sayi' adlı bir sınıf ile temsil edilecektir. Daha sonra, başka bir bağlı liste sınıfı oluşturulacak. Bu bağlı listede, her bir düğüm bir sayı listesini içerecek şekilde tasarlanacak ve bu düğümler 'Dugum' adlı bir sınıf içinde tanımlanacaktır. Bu yeni bağlı listenin sınıf adı 'SayilarListesi' olacaktır. Son olarak, program kullanıcıya çeşitli seçenekler sunacak bir menüyü ekrana getirecek ve kullanıcının seçimine göre ilgili işlemleri gerçekleştirecektir.

Ödevde Yaptıklarım

Öncelikle, Basamak ve Dugum sınıflarının yapıcı fonksiyonlarını inceleyelim. Bu fonksiyonlar, sınıf nesnelerinin oluşturulması sırasında otomatik olarak çağrılacak ve nesnelerin veri üyelerini başlatacaklar. Basamak sınıfının yapıcı fonksiyonu, parametre olarak verilen basamak değerini nesnenin değerine atayacak ve sonraki işaretçisini NULL olarak belirleyecek. Dugum sınıfının yapıcı fonksiyonu ise, parametre olarak verilen sayı işaretçisini nesnenin değerine atayacak ve sonraki işaretçisini NULL olarak belirleyecek. Sayi sınıfının ekleBasamak fonksiyonunu ise, bir sayıya yeni bir basamak eklemek için kullandım. Fonksiyon, parametre olarak verilen yeniBasamakDeğeri adlı değişkeni kullanarak yeni bir Basamak nesnesi oluşturacak ve bu yeni basamak nesnesinin sonraki işaretçisini, Sayi nesnesinin ilk basamağına bağlayacak. Sonra, Sayi nesnesinin ilk basamağını, yeniBasamak nesnesine güncelleyecek. Bu şekilde, sayıya yeni bir basamak eklemiş oldum. Son olarak, SayilarListesi sınıfının sayiEkle fonksiyonu ile bir sayı listesine yeni bir sayı ekledim. Fonksiyon, parametre olarak verilen yeniSayi işaretçisini kullanarak yeni bir Dugum nesnesi oluşturdu. Eğer sayı listesi boşsa, yeni düğümü hem listenin başı hem de sonu olarak belirledim. Eğer sayı listesi boş değilse, listenin sonundaki düğümün sonraki işaretçisini yeni düğüme bağlayacak ve yeni düğümü listenin sonu olarak güncelleyecek. Sayı listesine yeni bir sayıyı bu şekilde ekledim.

Sayi sınıfının tekBasamaklarıBasaAl fonksiyonunu, bir sayının tek basamaklarını başa almak için tasarladım. Oluşturduğum döngüde, simdikiDugum işaretçisi sayının basamaklarını gösterirken, önceki işaretçisi de bir önceki basamağı gösterir. Döngü, simdikiDugum işaretçisi NULL olana kadar devam eder. Eğer simdikiDugum->basamakDeğeri tek ise düğüm başa taşınacak.

Sayi sınıfının basamaklarıTersle fonksiyonunu, bir sayının basamaklarını ters çevirmek için tasarladım. Bu fonksiyonda, üç işaretçi kullandım: simdikiDugum, önceki ve sonraki. SimdikiDugum işaretçisi, sayının ilk basamağını gösteriyor. Önceki işaretçisi, simdikiDugum işaretçisinin gösterdiği basamağın solundaki basamağı gösteriyor. Sonraki işaretçisi ise simdikiDugum işaretçisinin gösterdiği basamağın sağındaki basamağı gösteriyor. Sonraki işaretçisini simdikiDugum işaretçisinin sonraki işaretçisine eşitleyerek, sonraki işaretçisinin sonraki basamağı göstermesini sağladım. Daha sonra SimdikiDugum işaretçisinin sonraki işaretçisini önceki işaretçisine eşitledim. Böylece, simdikiDugum işaretçisinin gösterdiği basamak ters yönde bağlandı. Ondan sonra Önceki işaretçisini simdikiDugum işaretçisine eşitledim ve önceki işaretçisinin bir önceki basamağı göstermesini sağladım. Son olarak SimdikiDugum işaretçisini sonraki işaretçisine eşitledim. Böylece, simdikiDugum işaretçisi sonraki basamağı gösterdi. Bu fonksiyonu SayilarListesi sınıfının basamaklarıTersle fonksiyonu altında çağırarak tersleme işlemini gerçekleştirmiş oldum.

SayilarListesi sınıfının enBuyuguCikar fonksiyonunu, bir sayı listesindeki en büyük sayıyı bulup listeden silmesi için tasarladım. Eğer sayı listesi boş değilse, en büyük sayıyı bulmak için bir döngü yazdım. Bu döngüde, enBuyukDugum ve simdikiDugum adında iki Dugum işaretçisi kullandım.

enBuyukDugum işaretçisi en büyük sayıyı, simdikiDugum işaretçisi de şu anki sayıyı gösteriyor. Döngü, simdikiDugum işaretçisi NULL olana kadar devam edecek. Eğer enBuyukDugum işaretçisi NULL ise veya simdikiDugum işaretçisinin gösterdiği sayının tam sayı değeri enBuyukDugum işaretçisinin gösterdiği sayının tam sayı değerinden büyük ise, bu durumda enBuyukDugum işaretçisini simdikiDugum işaretçisine eşitledim. Böylece, enBuyukDugum işaretçisi en büyük sayıyı gösterdi. Döngü bittiğinde enBuyukDugum işaretçisi sayı listesindeki en büyük sayıyı gösterecek.

Tanımladığım yıkıcı metotlarda da hafızadan alınan alanları serbest bırakarak, çöp oluşmasını önledim.

Ödevde Zorlandığım Yerler ve Eksik Kalan Kısımlar

Ödevimde SayilarListesi sınıfı içerisinde Sayi nesnelerinin bir bağlı liste olarak tutulması kısmını anlamakta zorlandım. Bu konsepti kavramak için çok araştırma yaptım ve örnek kodlar inceledim. Ayrıca tek basamaklı sayıları başa almak için gerekli algoritmayı uygulamakta da zorluk çektim. Ancak proje sürecinde bu problemleri aşarak hem bağlı listeleri hem de liste içerisindeki düğümlerin yer değiştirmesi işlemini öğrenmiş oldum.

Ödevimi başarıyla bitirdiğimi ve istenen bütün işlemleri doğru şekilde yaptığımı düşünüyorum. Ödevde yaptıklarımı ve karşılaştığım zorlukları da raporuma ekledim. Umarım ödevim sizin de istediğiniz gibi olmuştur.

Ayşenur ÖZKAN

Y225012153