Veri Yapıları 2. Ödev Raporu

Zeynep ALDİNÇ

B18121390

Dugum.hpp

Dugum classı içerisinde dosyadan almak istediğimiz kelimenin ekleneceği, string türündeki eleman ismindeki değer oluşturuldu. Kelime ilk kez geldiğinde yanına sıfır değeri, tekrar geldiğinde ise ilk geldiği düğümle arasındaki mesafeyi yazmak için kullanılacak değer ise int türünde sayi ismiyle oluşturuldu. Kelimenin bulunduğu düğüm ve tekrar ettiği düğüm arasındaki mesafenin bulunabilmesi için her düğüm için numara adında int tipinde bir değişken oluşturuldu. Oluşturulan Dugum nesnesini ileri ve geri hareket ettirebilmek için (çift yönlü bağlı liste) ileri ve geri pointer değerleri oluşturuldu. Dugum classı için yapıcı ve yıkıcı metotlar oluşturuldu.

Dugum.cpp

Dugum yapıcı fonksiyonuna parametre olarak içerisine yazılacak eleman ve sayi değerleri, düğüme verilecek numara; geri ve ilerisinin bağlı olduğu yerleri belirtmek için ise yine hpp’de tanımlanmış olan geri ve ileri pointerları gönderildi. Yıkıcı fonksiyonda ise kullanıcıya düğümlerin yok edildiği gösterildi.

LinkedList

LinkedList classında ilk oluşturulan düğümün adresini gösteren basDugum adında bir pointer ve diğer düğümleri oluşturmaya yarayan geçici adında bir pointer oluşturuldu. Bağlı listenin boyutunu belirlemek için int tipinde size adında bir değişken oluşturuldu. Düğüm oluşturmak için kullanacağımız Dugum yapıcı fonksiyonuna numara değeri gönderilebilmesi için numara adında int tipinde bir değişken oluşturuldu. Düğümün içerisindeki kelimenin uzunluğunu atabileceğimiz bir uzunluk değeri oluşturuldu. Verilen cümle kelime kelime ayrılıp bir diziye atılacağı için dizinin her bir indisini belirtecek bir int i değişkeni oluşturuldu. Her bir kelimeyi alabilmek için string kelime değişkeni ve 100 elemanlı bir dizi oluşturuldu.

LinkedList yapıcı fonksiyonunda i değeri ve uzunluk sıfıra eşitlendi. Her düğüm 1den başlanarak numaralandırılacağı için numara 1e eşitlendi. DosyaOku fonksiyonu çağırıldı. basDugum adında Dugum türünde bir pointer oluşturularak aynı zamanda ilk düğüm oluşturulmuş oldu (içerisine dizinin i. Elemanı, eleman ilk kez gelmiş olacağı için sayı olarak sıfır ve numarası eklendi). basDugum, oluşturulmuş olan geçici pointerına atandı ve iki pointerda aynı düğümü (ilk düğüm) göstermiş oldu. Sadece bir düğüm bulunduğundan dolayı boyut (size) 1e eşitlendi. Okunan dosyadaki kelime sayısı (uzunluk) kadar dönen bir döngü içerisinde add fonksiyonu çağırıldı.

add fonksiyonu içerisinde basDugum, gezici isimli yeni bir pointera eşitlendi. basDugum ile aynı yeri(il düğüm) gösteren gecici’nin ilerisine yeni bir düğüm eklendi. O düğümün gerisi tekrar gecici’ye eşitlendi. geçicinin ilerisi, gecici’ye eşitlenerek geçici pointerının yeni düğümü göstermesi sağlandı. Gezici NULL’dan ve gecici’den farklı olduğu sürece devam eden bir döngü oluşturuldu. Eğer gezici’nin elemanı ve gecici’nin elemanı birbirinden farklıysa gezici ilerletildi. Değilse gecici’nin numarasından gezici’nin numarası çıkarıldı ve geçcici’nin gösterdiği düğüme sayı değeri olarak atandı. Böylelikle yeni gelen kelimenin tekrar edip etmediği kontrol edildi ve eğer tekrar ediyorsa kelimenin ilk kez yazıldığı düğümle arasındaki mesafe uzunluğu yeni düğüme sayı olarak yazılmış oldu. Yeni bir düğüm eklendiği için size 1 arttırıldı.

içerik.txt dosyasını okuyacak DosyaOku fonksiyonu oluşturuldu. Burada kelime değeri önce boş bırakıldı ve dosyadan bu kelime okundukça kelime diziye atandı ve uzunluk bir arttırıldı. Eğer dosya açılamadıysa “Dosya Acilamadi” uyarısı gösterildi.

Kelimeleri tekrar yazmamaktan kazanılan byte, int \_byte adında bir değişkende tutuldu. Dugum türünde gezici adında bir pointer oluşturuldu ve basDugume eşitlendi. For döngüsü içerisinde gezici, liste sonuna gelene kadar ilerletildi ve bütün liste taranmış oldu. Döngü içerisinde if koşuluyla gezicinin geldiği düğüm içerisinde eleman olup olmadığı kontrol edildi ve eğer yoksa ikinci bir gezici pointerı oluşturuldu (gezici2) ve ilk geziciye eşitlendi. Gezicinin içinde bulunan sayı kadar, gezici2 bağlı listede geri götürüldü ve gezici2’nin elemanının uzunluğundan 1 çıkarılarak \_byte değişkenine eklendi. (Boş olan düğümde de bir sayı değeri bulunduğu için kazanca (\_byte) eklenen byte sayısından 1 çıkarıldı). Ve sayı döndürüldü.

Remove fonksiyonunda düğümlerin tek tek silinme işlemi gerçekleştirildi. Bunun için Dugum türünde, eskiDugum adında bir pointer oluşturuldu ve basDugum, ona eşitlendi. basDugum bir ileri düğüme alındı ve o düğümün gerisi NULL yapıldı. Böylelikle ilk düğüm boşa çıkmış oldu. Ve eskiDugum silinerek o düğüm de silinmiş oldu.

Clear fonksiyonunda bağlı listenin boyutu (size) 1e eşit olana kadar remove fonksiyonu çalıştırıldı.

Yazdır fonksiyonunda bağlı liste ve elde edilen kazanç ekrana yazdırıldı.

LinkedList yıkıcı fonksiyonunda clear fonksiyonu çağırılarak düğümler teker teker silindi. En son diğer düğümlerden ayrı olarak oluşturulan basDugum de silinerek bütün düğümler ve pointerlar temizlenmiş oldu.

Test.cpp bölümünde main fonksiyonu içerisinde LinkedList classından bir nesne oluşturuldu ve o nesneden Yazdir fonksiyonu çağırıldı.