



SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİ YAPILARI DERSİ

ADI SOYADI.....: SEDANUR SARIKOÇ - MERVE SARI
ÖĞRENCİ NUMARASI.: B181210110 - B181210074
ÖĞRETİM TÜRÜ: 1C

RAPOR

Ödevde bizden istenilen içerik isimli dosyada olan metni boşluğa göre ayırıp her bir kelime için bir düğüm oluşturmak, eğer düğümlerde aynı kelimeler varsa bunun kaç düğüm önce olduğunu bulmak ve aynı olan kelimeler için ne kadar byte kazancı olduğunu hesaplamak ve bunları ekrana yazdırmak.

Öncelikle Dugum ve Dosya isimli iki farklı sınıf oluşturduk. Bu sınıfların başlık dosyalarına eklediğimiz metotların isimlerini ve alt alanları tanımladık. .cpp dosyalarında ise gövdeleriyle beraber metotları yazdık.

Dugum sınıfının .cpp dosyasında kurucu fonksiyonda değişkenlere ilk değer ataması yaptık. DugumEkle fonksiyonunda düğüm boş mu diye kontrol ettirdik. Düğüm boşsa ilk elemanı oluşturup head ve tail değerlerini atadık. Eğer düğüm boş değilse yeni bir düğüm oluşturup önceki düğüme ekledik. Bu eklediğimiz son elemanı da tail değişkenine atadık. Bu işlemleri yapmak için if else yapısını kullandık. Ve bu koşulların içinde son olarak oluşturduğumuz düğümlere DataKontrol fonksiyonunda, fonksiyona parametre olarak gönderdiğimiz data'nın düğümde kendinden önceki düğümlerin datasına eşit olup olmadığını kontrol ettik. Datalar eşit olmadığında tuttuğumuz sayacı bir arttırdık. DugumlerKontrol fonksiyonunda bütün düğümler için datalarının kontrolünü gerçekleştirdik. DugumleriYazdir fonksiyonunda önce düğümleri yazdırdık. Sonra aynı olan kelimeler için byte kazancını hesaplayıp yazdırdık. En sonda da düğümleri sildirdik. Yıkıcı fonksiyonda heap bellek bölgesinden aldığımız alanları iade ederek çöp oluşmasını engelledik.

Dosya sınıfının .hpp dosyasında public alanda Dugum sınıfından nesne oluşturduk. .cpp dosyasının DosyaOku fonksiyonunda txt dosyasının içeriğini okuttuk. Sonrasında bu dosyadaki metni boşluklara göre ayırdık ve her bir kelime için düğüm oluşturduk. DosyaYazdir fonksiyonunda Dugum sınıfında yazdığımız DugumleriYazdir fonksiyonunu çağırıp düğümleri yazdırdık. DugumSil fonksiyonu ile oluşturduğumuz düğümlere NULL atayıp sonrasında düğümleri sildik.

Test dosyasında Dosya sınıfından dosya nesnesi oluşturduk. Oluşturduğumuz bu dosya nesnesi için DosyaOku, DosyaYazdir ve DugumSil fonksiyonlarını çağırdık. Sonrasında dosya nesnesini sildirdik.

Ödevde ilk başladığımızda oluşturduğumuz düğümlere farklı adres atayamadık. Başlangıçta hepsinin adresi aynıydı. Düğümlere farklı adres atamada zorlandık. Çöpleri temizlemede zorlandık. Txt dosyasından aldığımız metni boşluğa göre ayırmada zorlandık. Düğümlerdeki dataların aynı veriye sahip olup olmadığını kontrol ederken zorlandık. Aynı olan kelimeler için sayaç tutmada zorlandık.

Dosyadan gelen metni boşluğa göre ayırmayı bunları farklı adreslerdeki düğümlere atamayı öğrendik. Çöp oluşmaması için heap bellek bölgesinden aldığımız adresleri nerede ve nasıl iade etmemiz gerektiğini öğrendik.

Ödevde eksik bıraktığımız yer olduğunu düşünmüyoruz. Umarız öyledir.