

## ÖDEV RAPORU

2021 Güz Dönemi Veri Yapıları Dersi 1.Ödevinde yaptığım çalışmalar şu şekildedir:

Ödevde çift yönlü bağlı listede ekleme ve çıkarma işlemlerini dosyadan okuyarak yapan bir program yazmamız istenmiştir.

Sona ekleme fonksiyonu sona düğüm ekler. İlk önce düğüm yapısında yeni bir düğüm oluşturur. Eğer listede eleman yoksa ilk başta tanımlanan root değişkenine eşitler. Daha sonra geçici bir pointer tutarak ilk elemanı yani root değişkenini işaret eder. While döngüsüyle ilk elemandan son elemana kadar gider. Döngü sonunda geçici tutulan pointer son elemanı işaret edeceği için başta oluşturduğu düğümü son elemana eşitler. Ve listenin eleman sayısını bir artırır.

Herhangi bir yere ekleme işleminde verilen elemanın nereye ekleneceği (yani index) ve elemanın verisine ulaşılmaktadır. İlk önce sona ekleme fonksiyonu yardımıyla sona boş bir düğüm tanımlar. Eğer verilen indexte eleman yoksa yani index sayısından daha az eleman varsa sona ekler. Index varsa son elemandan başlayarak indexe gelene kadar düğümlerin verilerini bir sonraki elemanın verisine kopyalar.

Sondan silme işleminde geçici bir pointer tutar ve pointer ilk elemanı işaret eder. Eğer eleman sayısı 1 ise ilk elemanı siler ve eleman sayısını 1 azaltır, ilk değerini 0' a eşitler. Eğer eleman sayısı 1'den fazlaysa while döngüsü yardımıyla son elemana ulaşır. Son elemanın önceki değerine gider ve o elemanın sonraki değerini son elemanın önceki değerine eşitler. Sonra geçiciyle işaret edilen düğüm silinir. Liste eleman sayısı bir azaltılır.

Herhangi bir yerden silme işleminde fonksiyon parametre olarak index değeri alır ve indexteki elemanı siler. Eğer verilen indexte eleman yoksa yani index sayısından daha az eleman varsa sondaki elemanı siler. Geçici bir pointer tutar ve pointer ilk elemanı işaret eder. For döngüsü yardımıyla ilk elemandan verilen indexteki elemana ulaşır. Geçici artık indexteki elemanı işaret etmektedir. Daha sonra while döngüsüyle geçiciden yani indexteki elemandan başlayarak son elemana kadar kendisinden bir sonraki elemanın verisini kendine kopyalar. Son elemanı silmeden önce sonuncu elemandan bir önceki elemanın sonraki değerini 0' a eşitler. Son elemanı siler ve eleman sayısını bir azaltır.

Yazdırma işleminde ilk elemandan başlayarak son elemana kadar düğümün verilerini yazdırır.

Dosyadan okuma işlemlerinde satırları tek tek okur. Eğer dosya açılırsa işlemlere başlar. Satırdaki ilk elemanı komut olarak tutar. E veya S olmasını if kontrol işlemleri ile kontrol ettim. Aynı zamanda büyük küçük harf hassasiyeti olmaması için tolower fonksiyonunu kullandım. Eğer dosyadan silme komutunu okursa satırın ikinci elemanından diyeze kadar olan kısmı index adlı değişkene atar. Daha sonra silme fonksiyonunu çağırır ve siler. Eğer dosyadan ekleme komutunu okursa; satırdaki 2. elemandan başlayarak diyeze kadar olan kısmı index adlı değişkene atar, diyezden bir sonraki karakterden başlayarak son paranteze kadar olan kısmı data adlı değişkene atar. Index ve data tutma işlemlerinde substring fonksiyonundan yararlandım. Diyez karakterini bulmak için de find fonksiyonundan yararlandım. Dosyayı okumayı bitirince listenin son halini yazdırır.

Programın ekran çıktısı bu şekildedir:

```
Veri.txt
1   E(0#Esra Aydin)
2   E(13#Ata Alemdar)
3   E(2#Ece Sahin)
4   E(3#Su Cakir)
5   S(8)
6   S(1)
7   S(3)
8   S(10)
9   E(17#Hande Eroglu)
10  E(1#Pelin Dunca)
11  S(0)
12  E(2#Emre Karadeniz)
13  E(3#Ahmet Kul)
14  S(1)
15  S(266184)
16  E(26#Beyza Erkan)
17  E(2#Sude Naz Durgun)
18  E(3#Nisa Karan)
19  S(85)
20  E(61#Zeynep Kurt)
21  E(0#Burak Erol)
22  S(1)
23  E(0412#Begum Nazli Colak)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Beyza\Desktop\DoublyLinkedList> mingw32-make all
g++ -I "./include" -c ./src/Node.cpp -o ./lib/Node.o
g++ -I "./include" -c ./src/DoublyLinkedList.cpp -o ./lib/DoublyLinkedList.o
g++ -I "./include" -c ./src/main.cpp -o ./lib/main.o
g++ -o ./bin/program ./lib/DoublyLinkedList.o ./lib/Node.o ./lib/main.o
./bin/program
Burak Erol <--> Beyza Erkan <--> Sude Naz Durgun <--> Zeynep Kurt <--> Begum Nazli Colak <--> NULL
PS C:\Users\Beyza\Desktop\DoublyLinkedList> |
```