

Ödevde istenilen, bağlı liste yapısını düğümlerle oluşturmak, fakat bunu adresleri sabit tutup verileri kaydırarak yapmaktır. Bu ödevi yaparken düğüm yapısını, pointer yapısını ve pointer'ları neden kullanmak gerektiğini, klasörleme sistemini, dosyayı terminalden derleme işlemini, header dosyası mantığını, header dosyasında belirtilen “tanımlamadıysan tanımla” mantığını, belleğin heap bölgesinde veri tutma işleminin avantajlarını öğrendim.

İlk olarak bir “Düğüm” sınıfı oluşturdum ve bu sınıfta kurucu fonksiyon, yıkıcı fonksiyon, önceki ve sonraki değeri tutacak iki tane pointer oluşturdum. Bu işlemten sonra “Bağlı Liste” sınıfı yaratıp önceden oluşturduğum “Düğüm” sınıfını bu sınıfa dahil ettim. “Bağlı Liste” sınıfında yapıcı fonksiyon, yıkıcı fonksiyon, ekleme, silme, yazdırma fonksiyonları, bağlı listenin uzunluğunu tutacak bir uzunluk değişkeni ve bu düğümlerden ilkinin tutacak bir “Düğüm” pointer'ı oluşturdum.

Ekleme fonksiyonu için, ilki hiç eleman olmaması durumunda, ikincisi belirtilen indis değerinin geçersiz olması, bulunmaması veya son elemanı göstermesi durumunda olmak üzere 2 adet istisna oluşturdum. Hiç eleman olmadığı durumda fonksiyon ilk eleman pointer'ını bu eklenen elemana eşitliyor. İndisin bağlı liste uzunluğundan fazla veya geçersiz olması durumunda ise sona direkt yeni düğümü istenilen veri ile ekliyor. Bu istisnalar haricinde fonksiyonun çalışması için, bağlı listenin sonuna yeni bir düğüm ekledim. Bu eklenen son düğümden istenen indise kadar gidecek bir while döngüsü ile verileri bir sonraki düğümlere kopyalayarak istenilen indise geldim. İstenilen indisteki elemanın verisini eklenmek istenen veriye eşitledim. Her eklenen düğümde eklenen düğümün önceki adresinin son elemanı, son elemanın sonraki adresinin yeni eklenen elemanı göstermesini sağladım. Adres değişikliği yapılmadan verileri öteleyerek eleman eklemiş oldum.

Silme fonksiyonu için, ilki hiç eleman olmaması durumunda, ikincisi listede tek eleman olması durumunda, üçüncüsü belirtilen indis değerinin geçersiz olması, bulunmaması veya son elemanı göstermesi durumunda olmak üzere 3 adet istisna oluşturdum. Hiç eleman olmadığı durumda bir işlem yapılmıyor. Bir eleman olduğu durumda bu eleman silinip ilk düğümü tutan pointer değeri sıfırlanıyor. İndisin bağlı liste uzunluğundan fazla veya geçersiz olması durumunda ise direkt sondan bir düğüm siliniyor. Bu istisnalar haricinde fonksiyonun çalışması için, istenilen indisten son indise kadar gidecek bir while döngüsü ile verileri bir önceki düğümdeki verilere kopyalayarak son indise geldim. Son düğümü silerek silme işlemini tamamladım. Her silinen düğümde bir önceki düğümün sonraki değerini sıfırladım. Adres değişikliği yapılmadan verileri öteleyerek eleman silmiş oldum.

Son olarak main fonksiyonunda dosya okuma işlemi yaptım. “getline” fonksiyonunu kullanarak “Veri.txt” dosyasını satır satır okuttum. Okunan satırın ‘E’ olması ve olmaması şeklinde şartlı olarak kontrol ettirdim. E ise ekleme işlemi için for döngüsü ile satırdaki elemanları döndürdüm. ‘(’ karakterinden ‘#’ karakterine görene kadar geçici bir string değişkenine atadım. Bu değişkeni stoi fonksiyonu ile indis formuna getirdim. ‘#’ karakterinden ‘)’ karakterine kadar okutup geçici stringime atadım. Bu değeri ise veri stringime eşitledim. Elde ettiğim parametrelerle ekle fonksiyonumu çalıştırdım. Satırın başının ‘E’ olmaması durumunda ‘(’ karakterinden ‘)’ karakterine kadar okutup stoi fonksiyonu ile indis formuna çevirdim. İndis parametresini vererek silme fonksiyonunu çalıştırdım.