Ödevde bir derleyicinin bilgisayara analaması için kodları bilgisayarın anlayacağı hale getirmesi gibi biz de Veri.txt'deki komutları derleyicinin anlıyacağı hale getirmemiz istendi.

Bunun yanında bu komutlar dahilinde eklenen verileri iki yönlü bağlı listeye eklememiz gerekti. Ver.txt'deki verileri ifstream komutuyla açtım ardından oradaki verilerin indis ve veri kısımlarını ayırmak için textislemler sınıfını oluşturdum Burda komutOkuma adında bir fonksiyon tanımladım bu fonksiyonda öncelikle substr komutu kullanarak ekle ile sil komutları arasından hangisinin çağırıldığını tespit edip ardından ilgili komutu çağırdım. Ekle ve sil komutundan önce bu komutların işleyişini sağlayan Dugum ve BagliListe sınıflarından bahsedeceğim.

Dugum sınıfının header kısmında düğümün verisini ,sırasını ,yapıcı ve yıkıcı fonksiyonunu tanımladım .Düğümün önceki ile sonraki düğümlerinin konumlarını pointer kullanarak tanımladım . Dugum.cpp de ise önceki ve sonraki düğüm pointerlarını 0 ladım ve yıkıcı fonksiyonu yazdım.

BagliListe'nin private kısmında ilk değişkenini tanımaldım ardından public kısmında yazdığım getIlk komutuyla ise bu değişkenin farklı kaynak kodlardanda erişilebilinir hale getirdim. BagliListe'nin public kısmında sil , ekle yazdır metodlarını yazdım. Ve verilerin ekleneceği ve sileneceği düğümün sırasını belirtmek kullanıcıya kaldığı için overloading yöntemini kullandım. Bagliliste.cpp'de sırası belirtilmemiş ekle komutunun fonksiyonu veriyi en sona eklemek olduğu için while döngüsüyle listenin sonuna gidip yeni bir düğüm ekliyor. Sırası belirtilmemiş sil komutuda aynı mantıkla çalışıyor aralarındaki tek fark birnin dugum eklerken diğerinin silmesi. Sıralı sınıflarda ise bu komutlar iyice farklılaştı ekle komutunda Dugum sınıfndatanımlamış olduğum sıra değişkenini kullandım. While döngüsüyle önce tüm düğümlerin sırasını belirledim bunu yaparken gec değişkenini kullandım işlemin sonunda gec değişkeninden sonra gelen bir ekdüğüm oluşturup o düğümden itibaren gec'in sırası kullanıcı tarfından belirlenen sıraya gelene kadar veriyi sonraki düğüme kopyalayıp önceki düğüme giden bir while döngüsü tanımladım. Sıra belirtilmiş sil komutunda ise for döngüsüyle istenilen düğüme gitmesini sağladım ardından yine bir bu sefer ekle fonksiyonun aksi olarark bu sefer veriyi önceki düğüme kopyalayıp sonraki düğüme geçen while döngüsü tanımladım. Tabi bu komutları yazarken istisnayi durumların sorun çıkarmasını engellemek için çeşitli if komutları kullandım örnek olarak içinde düğüm bulunmayan bir listede sil komutu çalışamayacağından fonksiyon bunu algılamasını ver kendini durdurmasını sağladım.

Son olarak main.cpp de TextIslemleri için yeni bir değişken oluşturup bu ddeğiişkenelde yazdırma ve komutokuma fonksiyonlarını çağırdım.

Makefile'ın içinde de dosyalarının yönlerini çeşitli değişkenlere atayıp bu sayede dosyaları gerekli klosörlere dahil ettim. Bunun dışında komutun sonunda programı çalıştırdım. Makefile'da sadece ilk eylemi gerçekleştirdiği için all adında bir değişkenle iki fonksiyonuda kapsayarak Makefile'ın çalışmasını sağladım. Ve ödevde bu kısımda çok zorlandım gerek boşluk yerine tab koymamız gerekmesi olsun gerek headerlardaki hpp lerin kütüphanelerden önce tanımlanması gerekmesi ve daha niceleri. Ama hata sayısı kodu yazarken pek hoş hissettirmesede . Bu hatalar sayesinde ileride daha rahat kod yazıcağımı bilmek güzel.

Ali Said Sarıtemur G201210044 2B