**YZM 2124**

Veri Yapıları

2022-2023 Bahar Dönemi

**ÖDEV # RAPORU**

Teslim Tarihi: 14/05/2023

Öğrenci Adı: Ayşe Nur İBİŞ

Öğrenci Numarası: 212802029

1. Giriş

Ödevin amacını, ödev için geliştirilen programa giriş olarak verilen ve çıkış olarak alınan verileri belirtiniz. Kullanılan dosya ve metot sayısı birden fazla ise her birinin görevini belirtiniz.

Ödevin amacı, bir kargo şubesine gelmiş olan 120 kişinin işlemleri boyunca geçirmiş oldukları işlem süresi ve müşteri ID’lerini, Veri yapıları dersi içerisinde öğrenilen materyalleri de kullanarak, şubeye gelen müşterilerin verilmiş olan bilgilerini bir takım veri biçimleri kullanıp içerisinde tutarak gerçeklemek ve pratik kazanmaktır.

Bu programda, struct yapısı içerisinde tutulmuş olup giriş olarak verilen veriler; gelen 120 müşterinin ID ve işlem sürelerini tutacak olan parametreler, yığın veri yapısında kullanılmak üzere yine struct’ larda tutulan veri, kuyrukboyutu, müşteriler[ ], bas, son parametreleri ve bas, dugum, ptr işaretçileridir. Diğer giriş verileri ise tüm müşterilerin toplam işlem sürelerini ve ortalama iş sürelerini barındıracak olan verilerdir. Verilen giriş verileri neticesinde programdan çıktı olarak alınan veriler ise müşterilerin her birinin sahip olduğu ID numarası ve işlem süresinin çıktısı, geçen toplam işlem süresi ve müşterilerin ortalama işlem süreleri ve FIFO kuyruk yapısına göre kimlerin kuyrukta daha fazla beklediğini gösterecek şekilde büyükten küçüğe sıralanmış halini veren çıktı verileri olmaktadır.

Program İçerisinde Kullanılmış Olan Fonksiyonlar:

1. musteriBilgi() : Musteri yapısında verilmiş olan parametreler üzerinde aritmetik işlem yapmak üzere aşırı yüklenerek oluşturulmuş bir fonksiyondur.
2. Struct ListKuyruk yapısında bulunan:

* void kapat() : Dugum yapısından bir p işaretçisi oluşturup bas işaretçisi NULL değerini gösterene değin yığından eleman silip p işaretçisini başa doğru kaydırma işlemini gerçekleştirir.
* void ekle(int islemsure) : Gelen her bir müşterinin niteliğini(işlemsüresi) sırasıyla yığın yapısındaki düğümlere kaydetme işlemini gerçekleştirir.
* İnt cikar() : Gelen her bir müşterinin niteliğini(işlemsüresi) sırasıyla yığın yapısındaki düğümlerden silip verileri kuyruk yapısına eklemek üzere kuyrugaEkle() fonksiyonuna gönderir.
* Void kuyrugaEkle(int islemsure) : Dugum yapısından “yeni” adında yeni bir yığın oluşturup ilk oluşturulan yığını yeni oluşturulan yığın yardımıyla ters çevirerek gelen verileri kuyruk yapısına ekler.
* void kuyruktanCikar() : kuyrugaEkle() fonksiyonunda kuyruğa eklenmiş olan verileri işaretçiler yardımıyla teker teker çıkarıp aynı zamanda çıkarırken de verileri yazdırıp toplam işlem süresini hesaplama işini gerçekleştirir.

1. Struct DiziKuyruk yapısında bulunan:

* DiziKuyruk() : Fonksiyonu ile kuyruğun bas ve son işaretçilerine başta boş olduğunu programa iletmek üzere -1 değeri atanmaktadır.
* Bool bosmu() ve bool dolumu() fonksiyonları kuyruğun bas ve son elemanlarının dolu mu boş mu olduğunun kontrolünü yapıp döndürür.
* void kuyrugaEkle() : Kuyruğun bos ya da doluluk durumunun kontrolünü sağlayarak musteriler[ ] dizisine gelmiş olan verileri ters çevirerek kuyruğa aktarma işlemini gerçekleştirir.
* İnt kuyruktanCikar() : Fonksiyonu musteriler[ ] dizisinden alıp kuyruğa eklediği verileri tek tek geri kuyruktan çıkartma işlemini yapmaktadır.
* void yazdir() : Fonksiyon içerisinde kuyruktaki müşteri bilgileri ve FIFO kuyruk yapısına göre kimlerin kuyrukta daha fazla beklediğini gösterecek şekilde büyükten küçüğe sıralanmış kuyruk yapısının yazdırılması gerçeklenmektedir.

1. Geliştirme ve Çalıştırma Ortamları

Ödev Visual Studio 2022 içerisinde C++’ ın konsol uygulaması geliştirme eklentisi ile geliştirilip komut istemi üzerinde GNU derleyici ile derlenmiştir.

1. Veri Yapıları ve Değişkenler

Ödev geliştirilirken öncelikle programa girdi olarak verilecek olan müşteri nitelikleri array ve list yapısı şeklinde iki ayrı struct yapı içerisinde stack yapılarına aktarılıp ardından stack’te tutulan veriler array ve list ile oluşturulan stack’ler ayrı ayrı olmak üzere queue yapısına gerçeklenmiştir. Bu doğrusal yapıların(dizi, bağlı liste) program içerisinde kullanım sebebi aynı tip veriler kümesinin düzenli bir şekilde tutulmasına olanak sağlamalarıdır. Array ya da list yapısında sırasına göre yeni veri ekleme, istenen bir veriyi çıkarma, istenen veriyi yazdırma gibi işlemleri ise stack ve queue yapılarıyla yapmak kolay ve mümkün olduğundan dolayı ödev geliştirme aşamasında kullanımı uygun görülmüştür.

1. Program Akışı

Bu program bir queue veri yapısının iki farklı uygulamasını kullanarak(array, list) bir kargo şubesi müşteri hizmetleri sistemini simüle eder. Her müşteri için işlem sürelerini rastgele oluşturur, bunları kuyruklara ekler ve ardından her müşteri için işlem süresini ve toplam iş süresini takip ederek verileri işler.

Program, her müşteri için işlem süresi ve ID niteliklerini depolamak için “Musteri” yapısı tanımlar. Ayrıca list yapısında kullanılmak üzere “Dugum” ve stack yapılarını uygulamak için sırasıyla list ve array kullanan “ListKuyruk” ve “DiziKuyruk” olmak üzere farklı iki queue sınıfı tanımlar.

Programın ana işlevi; önce rastgele sayı üretecini başlatır, ardından iki queue sınıfını oluşturur. Daha sonra rastgele oluşturulmuş olan işlem sürelerini önce müşteri sayısınca(N) array ve list ile oluşturulan stack yapılarına daha sonra stack’ ten aldığı verileri queue yapısına tanımlar. Her müşteri için işlem süresini ve toplam iş sürelerini takip ederken döngü kullanarak queue yapısına işler. Son olarak her müşteri için ID, işlem süreleri, toplam işlem süreleri ve müşterilerin ortalama işlem süreleri hakkında bilgi verir.

1. Sonuç

Array yapısı aracılığı ile queue yapısına veri aktarımında boyut sınırlaması gelmesine karşın, list yapısı ile veri aktarımında boyut sınırlaması gelmediği için list yapısının avantaj sağladığını ve istenilen durumda stack ve queue yapılarına veri ekleyip çıkarabilme, istenilen biçimde yazdırabilme gibi kolaylıkların bulunduğunu ayrıca bellek yönetimini daha verimli hale getiren yapılar olduğu bu ödevin gerçeklenmesi neticesinde öğrenilmiş ve anlaşılmıştır.