FIT 2 [INF224] – 2017/18

Veri Yapilari ve Algoritmalar – TP 05

Gulfem Isiklar Alptekin – Ozgun Pinarer – Emre Dogan

Amac

Bu TP'de bir bankaya gelen kişilerin işlem görmesi için bir kuyruk uygulaması yapacağız. İşlem görecek kişiler 2 tip olarak tanımlanmıştır. Eğer sıraya gelen kişi zaten bankanın müşterisi ise sıranın direk en başına geçer. Ama bankanın müşterisi değilse, sıranın en sonuna yerleşir ve kendisine sıra gelene kadar işlem göremez.

1) Bu bankada tek bir gişe olduğunu varsayıyoruz. Dolayısıyla banka kuyruğundan da çıkışlar ancak bu tek gişe üzerinden yapılır.

2) Bankaya gelen kişilerin hepsinin sadece para yatırma işlemi yapmak üzere bankaya geldiğini varsayıyoruz.

3) Yeni gelen bir banka müşterisinin sırada başka banka müşterileri varsa bile yine de en öne geçtiğini varsayıyoruz.

Soru 1

Bankaya gelen kişileri tanımlayan `struct kisi{};` yapısını C dilinde yazınız. Bu yapı gelen kişinin müşteri olup olmadığı ve bankaya yatıracağı para değerini içerecektir.

Soru 2

Bankaya gelen kişilerin oluşturduğu kuyruk yapısını tanımlayınız. Bu kuyruğa ait `add()`, `remove()`, `create()`, `is_full()`, `is_empty()`, `display_all()` fonksiyonlarını yazınız.

Soru 3

Yukarıda yazmış olduğunuz klasik `add()` fonksiyonu gelen kişi banka müşterisi değilse kuyruğun sonuna o kişiyi ekler.
Kuyruktaki tüm elemanları birer hane geriye kaydıran `shift_all()` fonksiyonunu yazınız. Bu fonksiyon sonucunda kuyruğun en başında 1 kişilik boş yer açılmalıdır. Eğer kuyruk kaydırılamayacak kadar doluysa, bu fonksiyon hata döndürecektir. Yazdığınız `shift_all()` fonksiyonunu kullanarak, eğer bankaya gelen kişi müşteri ise o kişiyi kuyruğun en önüne ekleyen `add_customer()` fonksiyonunu yazınız.

Soru 4

Yazdığınız fonksiyonları test etmek için `main()` fonksiyonunu yazınız. `main()`'de şu işlemleri yaptırabilirsiniz.

- Boyutu 10 olan boş bir kuyruk yaratın
- Son kullanıcıdan alınan verilerle sırayla 5 tane kuyruk elemanı yaratıp bunları kuyruğa ekleyiniz. Bu kuyruk elemanlarının tamamı banka müşterisi olmayan normal kişilerdir.
- 2 adet banka müşterisi kişiyi kuyruğa ekleyin.
- 4 adet müşteri olmayan yeni kişiyi kuyruğa ekleyin (Kuyruk doluysa hata verecek unutmayın)
- Kuyruktan önce 2 kişi çıkarın. Sonra 1 tane banka müşterisi olmayan kişi ekleyin
- Kuyruğa 2 tane banka müşterisi ekleyin

Bu örnekte kişiler banka müşterisi olsalar bile kendilerinden sonra bir başka banka müşterisi geldiğinde onların önüne geçiyordu. Bu sorunu çözmek için tasarımı değiştiriyoruz. Bir yerine iki tane kuyruk yapısı kullanarak bu sorunu çözebiliriz.

Soru 5

Birinci kuyruk yapısı sadece müşteri olmayanların sırasını tutarken, ikinci kuyruk yapısı da sadece müşteri olanların sırasını tutacak şekilde gerekli yapıları kullanarak tanımlayınız (`struct queue_noncustomer` ve `struct queue_customer`).

Soru 6

Yeni bir `main()` fonksiyonunu yazınız. `main()`'de şu işlemleri yaptırabilirsiniz.

- Boyutu 5er olan boş birer müşteri ve müşteri olmayan kuyruklarını yaratın
- Bankaya öncelikle 4 tane müşteri olmayan gelir
- Ardından 2 tane müşteri olan gelir.
- Ardından 3 kişi bankada sırayla işlem yaptırır ve kuyruklarından çıkar. Ama burada, öncelikle müşteri olanların kuyruğu boşalacak, daha sonra müşteri olmayanların kuyruğu işlem görmeye başlayacaktır. Dolayısıyla müşteri kuyruğunda eleman kalmayana kadar, müşteriler sırasıyla banka işlemlerini yaptıracaklardır.

Onemli Notlar:

1. Yazdiginiz fonksiyonlara yorum satiri eklemeyi unutmayin. Yazdiginiz dongu ve kontrolleri aciklayan yorum satirlarini ekleyiniz. Yorum satiri eklenmemis odevler ve Tpler gecersiz sayilacaktir.
2. Yazdiginiz fonksiyonlardan header .h dosyasi olusturunuz ve .c dosyasinda sadece main() fonksiyonunu birakiniz.
3. Odev teslimi sirasinda yuklenecek tek bir dosyaniz olmasi durumunda, dosyayi "OgrenciNo_IsimSoyisim_TPX.c" gibi isimlendirip, birden fazla dosyaniz olmasi durumda ise gerekli dosyalari zipleyip tek bir dosya haline getirip, ziplenmis dosyayi da "OgrenciNo_IsimSoyisim_TPX.tar.gz" gibi isimlendirerek sisteme yukleyiniz.