

Galatasaray Üniversitesi
Bilgisayar Mühendisliği Bölümü
INF224 - Veri Yapısı ve Algoritmalar

30 Ekim 2023

TP 04 - Kuyruk

Dikkat!

- Soruların açıklamalarını dikkatlice okuyup, cevaplarınızı sizden istenen şekilde teslim edin.
 - Dosya isimlerinde, değişken isimlerinde, vb. Türkçe veya özel karakter kullananlardan puan kıracağım.
 - **Programı derleme hatası tetikleyenler doğrudan sıfır alacaktır.** Ödevinizi göndermeden önce kodlarınızın çalıştığını mutlaka kontrol edin, eksik olsa da düzgün şekilde çalışan bir kod gönderin. Gerekirse hatalı kısımları yorum satırına alıp, bana not yazın.
 - Kopya durumunda başınıza ne geleceğine dair uyarı yapmama gerek yoktur herhalde...
-

Bu TP’de bir bankaya gelen kişilerin işlem görmesi için bir kuyruk uygulaması yapacağız. İşlem görecekt kişiler banka müşterisi olanlar ve banka müşterisi olmayanlar olmak üzere 2 ayrı tip olarak tanımlanmıştır.

Eğer sıraya gelen kişi bankanın müşterisi ise sıranın en başına geçer. Ama bankanın müşterisi değilse, sıranın en sonuna yerleşir ve kendisine sıra gelene kadar işlem göremez.

Soru 1:

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

Soru 2:

Bankaya gelen kişilerin oluşturduğu kuyruk yapısını tanımlayınız (`struct queue`). Kuyruğa gelen kişilerden müşteri olanlara öncelik tanımak için iki ayrı kuyruk tanımlayınız (`struct queue customer`, `struct queue non-customer`).

Bu kuyruk yapısına ait (`struct queue`):

1. `create()`
2. `add()`
3. `remove_person()`
4. `is_full()`
5. `is_empty()`
6. `display_all()`

fonksiyonlarını yazınız.

Soru 3:

Yazdığımız fonksiyonları test etmek için `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.

Soru 4:

Burada tasarımıımızı değiştireceğiz:

Bu bankada tek bir gişe olduğunu varsayıyoruz. Dolayısıyla banka kuyruğundan da çıkışlar ancak bu tek gişe üzerinden yapılır. Kişileri tek bir kuyrukta (**struct queue**) tutup işlem yapacağız.

Bankaya gelen kişilerden müşteri olanların öncelikli olarak işlem sağlayabilmesi için kuyruğun en başına eklenmeleri gerekmektedir.

Ancak, 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 açılacaktır.

Eğer kuyruk kaydırılamayacak kadar doluysa, bu fonksiyon hata döndürecektir.

Yazdığımız **shift_all()** fonksiyonunu kullanarak, eğer bankaya gelen kişi müşteri ise o kişiyi kuyruğun en önüne ekleyen **add_customer()** fonksiyonunu yazınız. (Yeni gelen bir banka müşterisinin sırada başka banka müşterileri varsa bile yine de en öne geçtiğini varsayıyoruz.)

Soru 5:

Yazdığımız fonksiyonları test etmek için **main()** fonksiyonunu yazınız. **main()**'de şu işlemleri yaptırabilirsiniz.

- Boyutu 10 olan boş bir kuyruk yaratın
- 5 tane banka müşterisi olmayan kişiyi kuyruğa ekleyin.
- 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.

Önemli Notlar:

- Sizlere verilmiş olan iskelet koddaki **TODO**'ları takip ederek ödevinizi yapınız.
- **test.c**, **queue.c**, **queue.h** dosyalarını zipleyiniz. Zip dosyanızın ismi **TP04_0grenciNo_AdSoyad.zip** şeklinde olmalıdır.
- Örnek ekran çıktısı ve ekrana yazılacak print formatlarını aşağıda bulabilirsiniz.

```
Input: 1
Output:
Using customer and non-customer queues:
-----
4 non-customers are added to non-customer queue:
Customer queue:
Queue is empty.
Non-customer queue:
Queue Status:
Is Client: No, Money: 100.00
Is Client: No, Money: 200.00
Is Client: No, Money: 300.00
Is Client: No, Money: 400.00

2 customers are added to customer queue:
Customer queue:
Queue Status:
Is Client: No, Money: 500.00
Is Client: No, Money: 600.00
Non-customer queue:
Queue Status:
Is Client: No, Money: 100.00
Is Client: No, Money: 200.00
Is Client: No, Money: 300.00
Is Client: No, Money: 400.00

3 people exit the queue (2 customer, 1 non-customer):
Customer queue:
Queue is empty.
Non-customer queue:
Queue Status:
Is Client: No, Money: 200.00
Is Client: No, Money: 300.00
Is Client: No, Money: 400.00
```

```
Input: 2
Output:
Using solely one queue:
-----
5 non-customers are added to queue:
Queue Status:
Is Client: No, Money: 700.00
Is Client: No, Money: 800.00
Is Client: No, Money: 800.00
Is Client: No, Money: 1050.00
Is Client: No, Money: 1100.00

2 customers are added to queue:
Queue Status:
Is Client: Yes, Money: 1000.00
Is Client: Yes, Money: 550.00
Is Client: No, Money: 700.00
Is Client: No, Money: 800.00
Is Client: No, Money: 800.00
Is Client: No, Money: 1050.00
Is Client: No, Money: 1100.00

Queue is full, failed to add person.
4 non-customers are added to queue:
Queue Status:
Is Client: Yes, Money: 1000.00
Is Client: Yes, Money: 550.00
Is Client: No, Money: 700.00
Is Client: No, Money: 800.00
Is Client: No, Money: 800.00
Is Client: No, Money: 1050.00
Is Client: No, Money: 1100.00
Is Client: No, Money: 12000.00
Is Client: No, Money: 1300.00
Is Client: No, Money: 1400.00

2 people are removed from the queue:
Queue Status:
Is Client: No, Money: 700.00
Is Client: No, Money: 800.00
Is Client: No, Money: 800.00
Is Client: No, Money: 1050.00
Is Client: No, Money: 1100.00
Is Client: No, Money: 12000.00
Is Client: No, Money: 1300.00
Is Client: No, Money: 1400.00

1 non-customers are added to queue:
Queue Status:
Is Client: No, Money: 700.00
Is Client: No, Money: 800.00
Is Client: No, Money: 800.00
Is Client: No, Money: 1050.00
Is Client: No, Money: 1100.00
Is Client: No, Money: 12000.00
Is Client: No, Money: 1300.00
Is Client: No, Money: 1400.00
Is Client: No, Money: 1600.00
```