

Veri Yapıları Dersi

2. Ödev [Tüm Şubeler İçin]

Teslim Tarihi

2 Aralık 2018 (Pazar günü saat 23:59'a kadar.)

Ödev İçeriği

Yazacağınız program C++ konsol uygulaması olup çalıştırıldığında Sayi.txt isimli dosyayı okuyup **Çift Yönlü Dairesel Bağlı Listeyi** oluşturmalıdır. Okuyacağı Sayi.txt dosyası formatı aşağıda verilmiştir.

159 12 1 9632 74 32 610 12 799 850

Her sayıyı bir sonraki sayıdan bir boşluk ayırmaktadır. Dosyada 10000 veya 100000 adet sayıda bulunabilir. Çift Yönlü Dairesel Bağlı Liste **her düğümünde bir yığıt** içermelidir. Program çalıştığında ekrana aşağıdaki menü gelmelidir.

- 1- Sola Hareket
- 2- Sağa Hareket
- 3- Bir Düğümün Durumlarını Yazdır
- 4- Tüm Durumları Yazdır
- 5- Çıkış

Çıkış seçilmediği sürece program devam etmelidir. Her sınıf ve veri yapısı **Heap bellek bölgesinde** oluşturulmalı ve **çöp oluşumu** engellenmelidir.

Yığıtın bu ödevdeki işlevi düğümün üzerinden geçen değerleri kaydetmesidir. Çift Yönlü Dairesel Bağlı Listenin birinci ve sonuncu düğümlerinin adresleri tutulmalıdır. Bu sola veya sağa hareket etmede kolaylık sağlayacaktır.

Menülerde Yapılacak İşlemler

- 1- Sola Hareket

Her düğümün içerisindeki sayı bir sol düğüme kaydırılacaktır. Düğümler aynı yerlerinde kalmalı sadece veriler bir sola ilerlemelidir.

- 2- Sağa Hareket

Her düğümün içerisindeki sayı bir sağ düğüme kaydırılacaktır. Düğümler aynı yerlerinde kalmalı sadece veriler bir sağa ilerlemelidir.

- 3- Bir Düğümün Durumlarını Yazdır

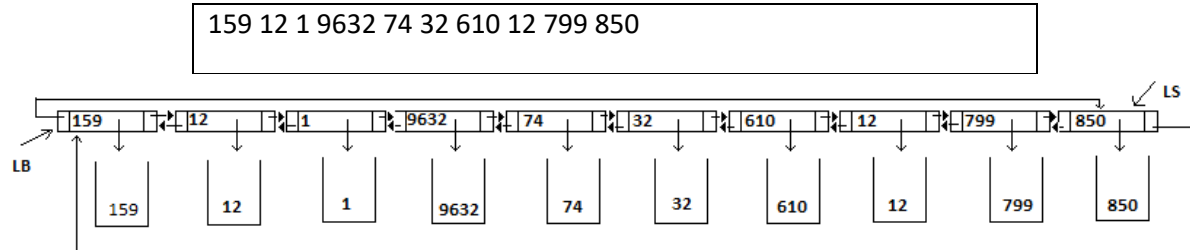
Bir düğümün üzerinden geçen sayıları yazdırabilmek için içerisinde bulunan yığıttan faydalanılacaktır. Yığıtın elemanları yazdırılacaktır. Menüden 3 seçildiğinde yığıt elemanları silinmemelidir. Sadece yazdırılmalıdır.

4- Tüm Durumları Yazdır

Menü 3 ile aynı yapıda çalışacaktır, fakat bütün düğümlerin üzerinden geçen sayıları yazdıracaktır.

Örnek Bir Senaryo

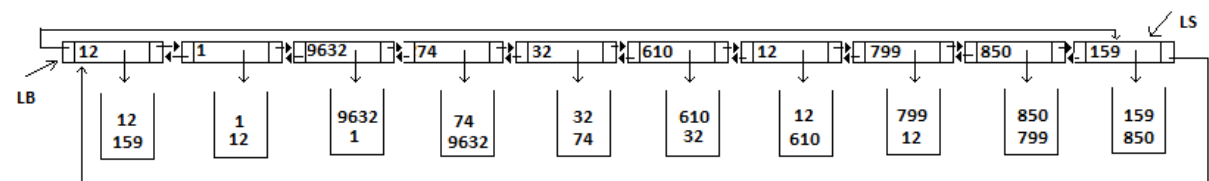
Sayi.txt dosya içeriği aşağıdaki gibi olursa...



İlk etapta kullanıcı 4. Seçeneği seçti ve tüm durumları ekrana yazdırdı. Aşağıdaki gibi bir ekran gelecektir. Herhangi bir hareket olmadığı için her düğümün yığtının içeriği düğümün içeriği ile aynı olacaktır. Yazılış biçimi --> Düğüm içerisindeki Sayı : İlgili düğümdeki Yığtın sayıları

159	:	159
12	:	12
1	:	1
9632	:	9632
74	:	74
32	:	32
610	:	610
12	:	12
799	:	799
850	:	850

Bu durumdayken kullanıcı sola hareket dedikten sonra 4. Seçeneği seçip ekrana yazdırırsa aşağıdaki gibi bir sonuç elde edecektir.



12	:	12	159
1	:	1	12
9632	:	9632	1
74	:	74	9632
32	:	32	74
610	:	610	32
12	:	12	610
799	:	799	12
850	:	850	799
159	:	159	850

Sağa hareket dediği zaman yukarıdakine benzer sadece verile sağa doğru hareket edecektir.

Bir düğümün durumunun yazdırılmasında kullanıcıdan düğümün index numarası istenecek ve ilgili indexteki düğümün verisi ve yığıtı ekrana yazdırılacaktır.

```
Dugum Konumu:7
799      :      799      12
```

Çok Önemli: Çift Yönlü Dairesel bağlı liste dizi ile gerçekleştirilemez ve kesinlikle şablon sınıf olamaz!!

Her düğüm kendi yığıtını içermeli ve Yığıt sınıfı şablon olmamalıdır!!

Sınıfların alt alanları private olmalı ve bu alt alanlara gerektiğinde erişim metotlar yardımıyla yapılmalıdır. Programda goto kullanılamaz. Sınıflar laf olsun diye tasarlanmamalı görevlerini yapmalıdırlar.

Önemli Not: Raporunuz detaylı olmalı ve kendi cümleleriniz olmalıdır. Kopya ödevler sıfır olarak değerlendirilecektir. SABİS şifreniz sizin sorumluluğunuz altındadır eğer arkadaşınız sizden habersiz ödevinizi alırsa bundan sizde sorumlu tutulur ve sıfır alırsınız.

Teslim Formatı

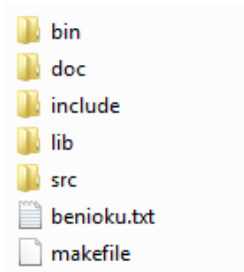
Yazacağınız tüm sınıfların başlık ve kaynak dosyaları ayrı olmalıdır. **Şablon sınıf veya Şablon fonksiyon kullanılamaz.**

Klasör Hiyerarşisi

- src Klasörü (Kaynak dosyalarınızın bulunduğu klasör)
- **lib Klasörü** (.o dosyalarının bulunduğu klasör)
- include Klasörü (Başlık dosyalarınızın bulunduğu klasör)
- doc Klasörü (Raporunuzun ve eğer varsa diğer dokümanların bulunduğu klasör.)
- bin Klasörü (Çalıştırılabilir dosyalarınızın (.exe) bulunduğu klasör.)
- benioku.txt (Program için ayrıca belirtmek istedikleriniz. Önerileriniz. Boş kalabilir.)
- **makefile**

Makefile olmayan ödev derlenmemiş olarak kabul edilmektedir (-50 puan).

Rapor Yoksa (-15 Puan)



Rapor pdf formatında olmalıdır. Raporu ayrıca çıktı olarak getirmenize gerek yoktur. Raporunuzda kısaca sizden istenilen, öğrendikleriniz, ödevde yaptıklarınız, eksik bıraktığınız yerler, zorlandığınız kısımlar anlatılabilir. **Ödev raporunda yazı boyutu 11 punto olmalıdır ve rapor en az 1 sayfa en çok 2 sayfa olabilir.**

Klasörlerinizi bir ana klasör içine koyarak .rar'layıp <http://www.csodev.sakarya.edu.tr> adresi üzerinden gönderiniz. Bu sistem üzerinden ödevin nasıl gönderileceği http://content.lms.sabis.sakarya.edu.tr/Uploads/50105/28982/odev_nasil_gonderilir.pdf dosyasında gösterilmektedir. Yukarıda belirtilen teslim tarihinden sonra gönderilen ödev **kesinlikle kabul edilmeyecektir.**

Yazmış olduğunuz bütün kaynak kodların en başında aşağıdaki bilgiler bulunmalıdır. Bilgileri kendinize göre güncelleyiniz.

```
/**
 * @file           Dosya adı
 * @description    Programınızın açıklaması ne yaptığını dair.
 * @course         Dersi aldığınız eğitim türü ve grup
 * @assignment     Kaçınıcı ödev olduğu
 * @date           Kodu oluşturduğunuz Tarih
 * @author         Gruptakilerin yazar adları ve mail adresleri
 */
```

Önemli: Yazacağınız kodlar GNU derleyicisinde (MinGW) kesinlikle derlenmelidir. Derlenmeyen kod itibar görmez ve çok düşük puan üzerinden değerlendirilir. Ödevi bireysel gönderebileceğiniz gibi grup olarak gönderebilirsiniz gruplar en fazla 2 kişiden (Herkes kendi şubesinden bir arkadaş ile grup kurabilir. 1A ve 2A şeklinde şubeler farklı şube kabul edilmektedir ve grup oluşturamazlar.) oluşabilir.

KOPYA ÖDEV SIFIR OLARAK DEĞERLENDİRİLMEKTEDİR*

* Şubeler farklı dahi olsa kopya ödev gönderenler sıfır alacaklardır.