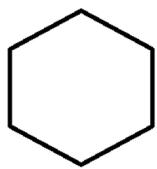
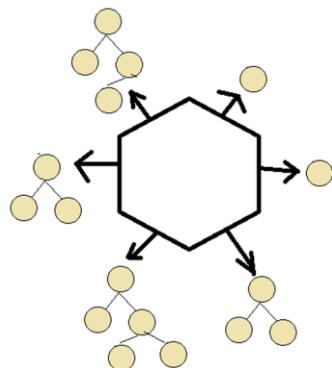


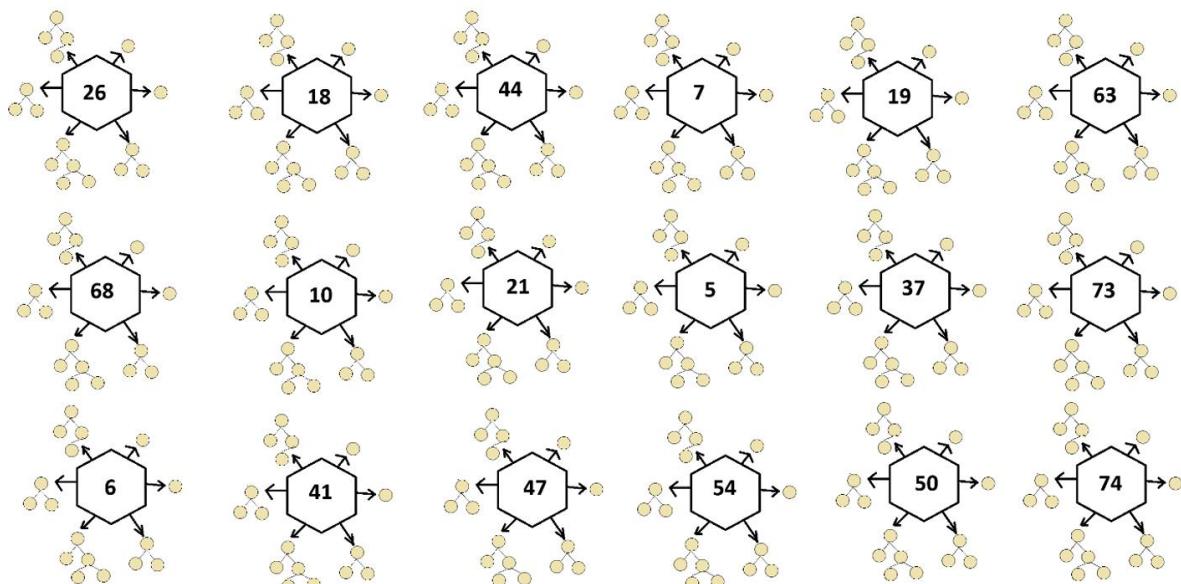
MinGW C++ ortamında derlenen bir program yazılacaktır. Bu ödevde aşağıda verilen altıgen şēkil, bir öncelikli kuyruk veri yapısını temsil etmektedir. Kuyruğa en fazla 6 veri eklenebilmektedir. Her kenar bir veri ve bu veri ikili arama ağacını tutmaktadır.



İkili arama ağaçlarında eşit değer gelirse sola eklenecektir.



Tüm yapı aşağıdakine benzer olacaktır. Buradaki sayılar kuyrukta çıkmak üzere olan ağaçın kökündeki değerin öncelikli çıkmak üzere olan ağaç kökündeki değere bölünmesi ile elde edilir. Öncelik ağaç yüksekliği ile belirlenir. Ağaç yüksekliği fazla olan önceliklidir. Örneğin çıkmak üzere olan bir ağaçın kökü 75, öncelikli çıkmak üzere olan ağaçın kökü 25 ise orada 3 değeri yazmalıdır. Sayı ve bölümler tam sayı olarak yapılmalıdır.  $75/25=3$



Kuyruk ve ağaçlar dosya okunarak oluşturulacaktır. Data.txt aşağıdaki gibi bir formatta olacaktır. Her satırındaki sayılar soldan sağa doğru ikili arama ağacına yerleştirilecek. Satır

bitince oluşan ikili arama ağacı kuyruğa eklenecektir. Altıgen yapı bitti mi sıradaki altıgen yapıya geçilecektir.

73 5 48

91 16 33 7 82

40

12 67 58 94

3 29 77 10 86 54

100 8

61 25 44 98 14

72 6 31 89 53 20 1

17 95

26 62 39 11 84

57 9 32 45 81 13 70 22

50 27 66 19 96 35

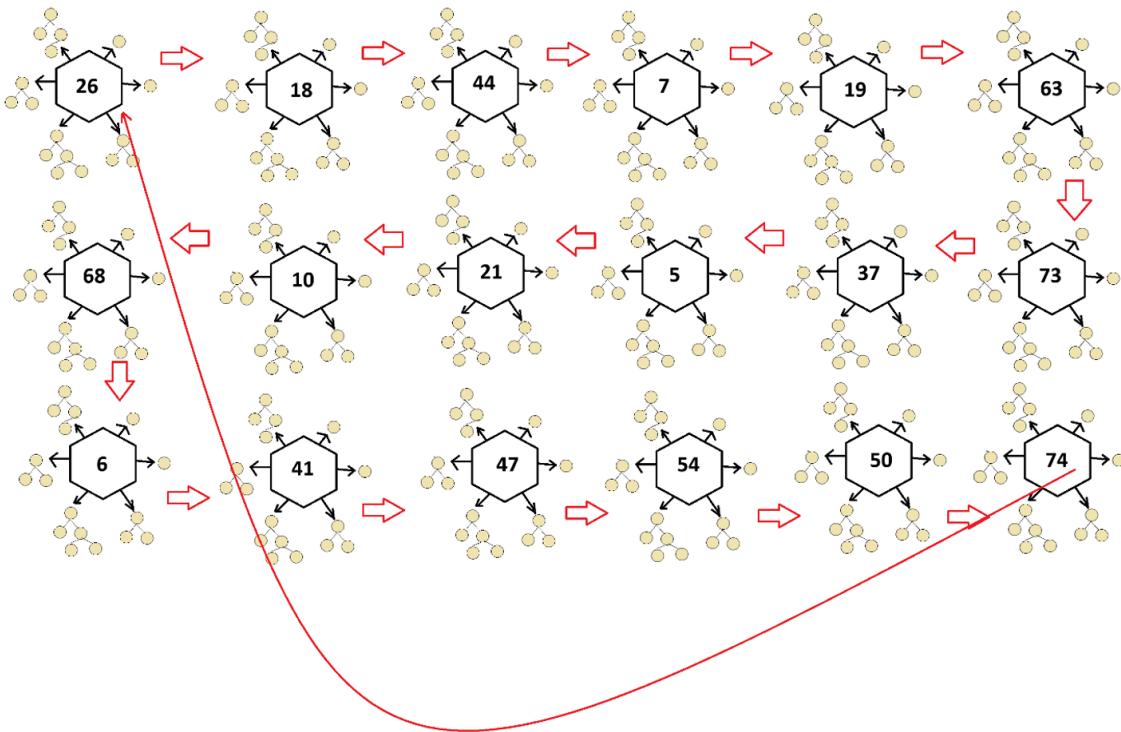
Dosya okunma aşamasında toplam kaç altıgen olabilir hesaplanıp ekranda boş olarak gösterilecek. Dosyadan satırlar okundukça ekran güncellenecek ve kuyruklardaki sayısal değişim gözlemlenebilecektir. Tüm dosya okunduktan sonra oluşan sayısal durum ekranda gösterilecektir. Ekranda her satırda en fazla 6 adet kuyruk gösterecektir. Sayılar anlaşılmasına için aşağıdaki sayılar rastgele ve temsili verilmiştir.

26	18	44	7	19	63
68	10	21	5	37	73
6	41	47	54	50	74

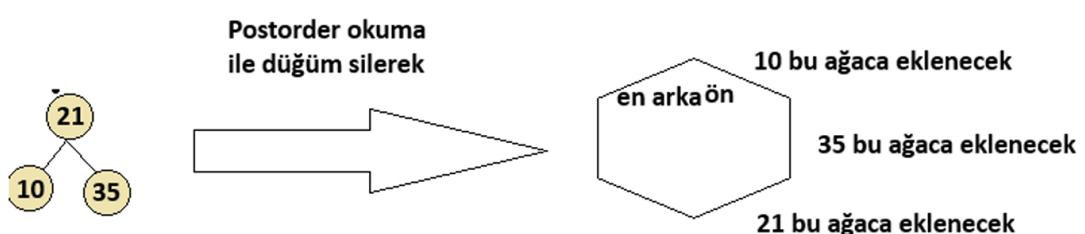
Kullanıcıdan kaç tur çalışacağı istendikten sonra bir tuşa bastığı gibi turlar başlayacak ve ekranda hangi turda olduğu görülecek ve sayıların da her turda son durumları ekranda görünecek yani ekran sürekli güncellenecektir. Ekranda daha önce yazılanlar temizlenmelidir. Turlar bitince sayıların son durumu ekranda görünmelidir. Ekran bu süre zarfında durdurulmayacak. Sadece son durum ekranda görünür duracaktır.

**Tek numaralı turlarda** kuyruklardan normal veri (ağaç) çıkarılacak bir sağındakı altigen yapıdaki ağaçlara eklenecektir. **Çift numaralı turlarda** kuyruklardan öncelikli veri (ağaç) çıkarılacak bir sağındakı altigen yapıdaki ağaçlara eklenecektir.

Ekleneceğinin yönü aşağıda verilmiştir.



Diğer altigene ekleme şu şekilde yapılacaktır. Bir ikili arama ağacı komple kuyruktan silinip altigene eklenirken postorder bir şekilde düğümler silip kuyrukta önden arakaya doğru her ağaca bir düğüm eklenecektir. Düğümler bitmez ama kuyruğun sonuna gelinirse tekrar aynı kuyruğun önünden başlanıp devam edilecektir.



Nesne yönelimli yaklaşım ile kod yazılmalıdır. Programın yavaş çalışması durumunda puan kırılacaktır. En az 50.000 satırlı bir Data.txt ile mutlaka deneme yapınız ve süreyi gözlemleyiniz.

**Turlar bittiğinde son durum ekranda gösterilmelidir.**

**Hiçbir şekilde Şablon (Generic) veri yapısı kullanılamaz. Her sınıfın başlık ve kaynak dosyası ayrı olmalı ve başlık dosyasında metot gövdesi bulunmamalıdır. Hazır veri yapısı kullanılamaz.** Önemli Not: Raporunuz detaylı olmalı ve kendi cümleleriniz olmalıdır. Kopya ödevler sıfır olarak değerlendirilecektir.

### **Teslim Formatı**

#### **Klasör Hiyerarşisi**

- src Klasörü (Kaynak dosyalarınızın bulunduğu klasör)
- **lib Klasörü** (.o dosyalarının bulunduğu klasör)
- include Klasörü (Başlık dosyalarınızın bulunduğu klasör)
- doc Klasörü (Raporunuzun 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 veya yetersiz ise (-15 Puan)**

**Önemli: Yazacağınız kodlar MinGW derleyicisinde kesinlikle derlenmelidir.  
Derlenmeyecek kod itibar görmez ve çok düşük puan üzerinden değerlendirilir.  
Sadece C++ kodları kullanılmalıdır.**