



Amac

İçerisinde integer sayı değerleri tutan düğümlerden oluşan bir ikili arama ağacı yaratıp bu ağaç üzerindeki operasyonları tanımlamak. Uygulama olarak telefon rehberi gerçekleştirilecektir. Telefon numaraları ülke kodlarına göre ağaç yapısına yerleştirilecektir. Örnek ülke kodları aşağıdaki gibidir:

Almanya	49	Belçika	32	Fransa	33	Türkiye	90
Arjantin	54	Mısır	20	Hollanda	31	İsviçre	41
Avustralya	61	Monaco	377	İngiltere	44	Yunanistan	30
Avusturya	43	Brezilya	55	İspanya	34		
Meksika	52	Portekiz	351	İsveç	46		

Soru 1

Yaratacağınız ikili arama ağacını betimleyen yapıyı (struct) C dilinde tanımlayınız.

Soru 2

Tanımladığımız yapıyı kullanarak arama ağacına, parametre olarak verilen bir integer sayıyı, ağaçtaki en uygun yere ekleyen insertNode fonksiyonunu yazınız. Öyle ki;

- Bu ağaç henüz boşsa, bu fonksiyon aldığı ilk değerle tek düğümlü bir ağaç yaratacak
- Bu ağaçta hali hazırda eleman varsa bu fonksiyon yeni gelen sayının ekleneceği düğümü bulup düzgün yere ekleme yapılacaktır. (Ekleme yerini bulmayı kolaylaştırmak için özyinelemeli çağırma kullanınız)

Soru 3

Ağaçtaki en küçük sayıyı taşıyan düğümü bulan `findMin` ve en büyük sayıyı taşıyan düğümü bulan `findMax` fonksiyonlarını yazınız.

Soru 4

Ağaçtan, parametre olarak verilen bir integer sayıyı silecek `deleteNode` fonksiyonunu yazınız. Öyle ki;

- Bu ağaç henüz boşsa, bu fonksiyon hata döndürecek
- Bu ağaçta hali hazırda eleman varsa bu fonksiyon silinecek sayının öncelikle ağaçtaki yerini bulacak, sonra silme işlemini yapacak. Ağaçta o sayı yoksa hata döndürecek (Sayının yerini bulmayı kolaylaştırmak için özyinelemeli çağırma kullanınız. Gerekiyorsa Soru 3'te yazdığınız `findMin` fonksiyonunu kullanarak silme işlemini yapınız.)

Soru 5

Ağaçta bulunan sayıları `preorder`, `inorder`, `postorder` ve `levelorder` sırasıyla sıralayıp ekrana basacak 4 fonksiyonu yazınız.

Soru 6

Main fonksiyonunu daha islevli hale getirmek için switch-case yapısı ile kullanıcı menüsü oluşturunuz.

AVL Agac Uygulamasi

Soru 7

Main fonksiyonu sırasıyla 10,20,30,40,50,25 değerlerini yarattığınız ağaca eklesin. Sonrasında ağaçtan 40,20 ve 25 değerlerini silip ekranda her bir silme işleminden sonra ağacı gösteriniz.

Bu işlemleri yapmak için ihtiyaç duyacağınız

```
struct Node *leftRotateAVL(struct Node *x)
```

```
struct Node *rightRotateAVL(struct Node *x)
```

```
struct Node* insertAVL(struct Node* node, int value)
```

```
struct treeNode *deleteNodeAVL(treeNode *node, int data) fonksiyonlarını yazınız.
```

Önemli Notlar:

1. TP çalışmalarını “header file – source file – test file” şeklinde gerçekleştirmeniz gerekiyor: Yazdığınız fonksiyonların headerlarını header dosyasına, fonksiyonların kodlarını source dosyasına, test kodlarını da test dosyasına yazınız.
2. Dosyalarınızı zipleyip “OgrenciNo_IsimSoyisim_TPX” şeklinde isimlendirerek sisteme yükleyiniz.
3. TP çalışmalarını linux ortamında, herhangi ek bir IDE’ye gerek duymadan (kate, gedit vs) yapılması tercihimizdir. Farklı platformlarda hazırlanan ödevlerin kontrolü sırasında dosya uzantısı vb. sebeplerden çalışmaması öğrencinin sorumluluğundadır.