



Sınavda Uyulması Gereken Kurallar

1. Cep telefonlarının, hesap makinesi, saate bakmak gibi herhangi bir amaçla kullanılması yasaktır. Telefon **kapalı** ve **cepte** olmalıdır.
2. Sorularda anlaşılmayan noktalar ilk **20 dakika** sorulabilir. Her öğrencinin **1 kez** soru sorma hakkı vardır.
3. Soru kağıdına numaranızı ve isminizi yazıp imzalayınız. Soru kağıdı **Sizde kalacaktır**.

NUMARA :

AD SOYAD :

İMZA :

```
void addBelowRootModified(Node* p, int elt)
{
    while(p->left != NULL && p->right != NULL)
    {
        if( p->elt > elt)
            p = p->left;
        else
            p = p->right;
    }

    Node* newNode = new Node;
    newNode->elt = elt;
    newNode->par = p;

    if (newNode->elt < p->elt)
        p->left = newNode;
    else
        p->right = newNode;

    n += 1;
}

void main()
{
    LinkedBinaryTree nuTree;

    nuTree.addRoot();
    nuTree._root->elt = 8;

    nuTree.addBelowRootModified (nuTree._root, 7);
    nuTree.addBelowRootModified (nuTree._root, 6);
    nuTree.addBelowRootModified (nuTree._root, 5);
    nuTree.addBelowRootModified (nuTree._root, 4);
    nuTree.addBelowRootModified (nuTree._root, 3);
    nuTree.addBelowRootModified (nuTree._root, 2);
    nuTree.addBelowRootModified (nuTree._root, 1);
    nuTree.addBelowRootModified (nuTree._root, 9);
    nuTree.addBelowRootModified (nuTree._root, 10);
    nuTree.addBelowRootModified (nuTree._root, 11);
    nuTree.addBelowRootModified (nuTree._root, 12);
    nuTree.addBelowRootModified (nuTree._root, 13);
    nuTree.addBelowRootModified (nuTree._root, 14);
    nuTree.addBelowRootModified (nuTree._root, 15);

    cout<< "Inorder Traversal : ";
    nuTree.inorder(nuTree._root);
}
```

1.

- a) Yukarıdaki programın çıktısı nedir? (15P)
- b) addBelowRootModified() isimli fonksiyon, veriler aşağıdaki sırada eklendiğinde doğru çalışırken yukarıda main() fonksiyonundaki gibi eklenince yanlış çalışmaktadır. Fonksiyondaki hatayı düzeltiniz. Cevabınızda fonksiyonun tamamını yazınız. (15P)

8 4 12 2 6 10 14 1 3 5 7 9 11 13 15

8 7 6 5 4 3 2 1 9 10 11 12 13 14 15

2.

- a) Yukarıdaki verileri ikili ağaca yerleştiriniz.
- b) İkili ağaçtaki verileri **inorder** kuralına göre yazınız.
- c) İkili ağaçtaki verileri **preorder** kuralına göre yazınız.
- d) İkili ağaçtaki verileri **postorder** kuralına göre yazınız.

Not → Verileri ağaca önce 8 (kök), sonra 7, sonra 6, sonra 5, ... en son 15 olarak yerleştiriniz. (40P)

```
void HeapPriorityQueue::insert(const int& e)
{
    T.addLast(e);
    Position v = T.last();
    while (!T.isRoot(v)) {
        Position u = T.parent(v);
        if (!isLess(*v, *u)) break;
        T.swap(v, u);
        v = u;
    }
}

void main()
{
    HeapPriorityQueue Heap;

    Heap.insert(8);
    Heap.insert(7);
    Heap.insert(6);
    Heap.insert(5);
    Heap.insert(4);
    Heap.insert(3);
    Heap.insert(2);
    Heap.insert(1);
    Heap.insert(9);
    Heap.insert(10);
    Heap.insert(11);
    Heap.insert(12);
    Heap.insert(13);
    Heap.insert(14);
    Heap.insert(15);

    cout << "Heap Elements after Insertions :";
    Heap.print();

    Heap.removeMin();

    cout << "Heap Elements after removeMin :";
    Heap.print();
}
```

3. Yukarıdaki programın çıktısı nedir? (30P)