


```

void LinkedBinaryTree::traverse(Node* p)
{
    while (root != NULL)
    {
        while (p->left != NULL) p = p->left;
        cout << p->elt << endl;
        deleteNode(root, p->elt);
        p = root;
    }
}

void main()
{
    // Çıktı
    LinkedBinaryTree Tree;
    Tree.addRoot();
    Tree.root->elt = 8;
    Tree.addBelowRoot(Tree.root, 4);
    Tree.addBelowRoot(Tree.root, 12);
    Tree.addBelowRoot(Tree.root, 2);
    Tree.addBelowRoot(Tree.root, 6);
    Tree.addBelowRoot(Tree.root, 10);
    Tree.addBelowRoot(Tree.root, 14);
    Tree.addBelowRoot(Tree.root, 1);
    Tree.addBelowRoot(Tree.root, 3);
    Tree.addBelowRoot(Tree.root, 5);
    Tree.addBelowRoot(Tree.root, 7);
    Tree.addBelowRoot(Tree.root, 9);
    Tree.addBelowRoot(Tree.root, 11);
    Tree.addBelowRoot(Tree.root, 13);
    Tree.addBelowRoot(Tree.root, 15);

    binaryTree.traverse(binaryTree.root);
}

```

```

void insertOrdered(string& e, int& i)
{
    CircularlyNode* newNode = new CircularlyNode;
    newNode->elem = e;
    newNode->score = i;

    if (cursor == NULL)
    {
        newNode->next = newNode;
        cursor = newNode;
        return;
    }

    CircularlyNode* front = cursor->next;
    CircularlyNode* back = cursor;

    while( newNode->score > front->score )
    {
        back = front;
        front = front->next;

        if (.....) break;
    }

    back->next = newNode;
    newNode->next = front;

    if (newNode->score > cursor->score)
        cursor = cursor->next;
}

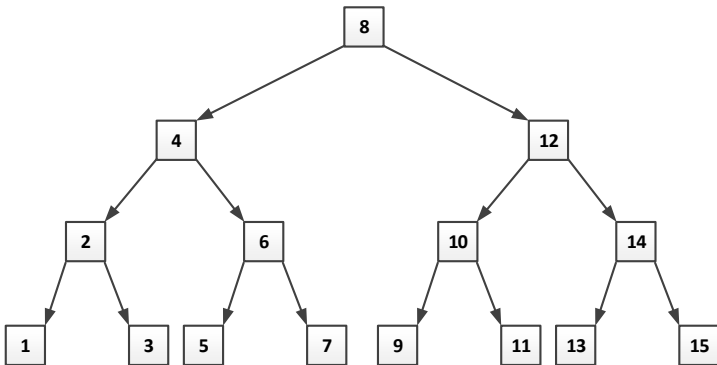
```

2. a) Yukarıdaki programın çıktısı nedir? (20P)

Not→ Silinen düğümün yerine kendisinden büyük en küçük düğümün geldiğini varsayınız.

b) Yukarıdaki programın çıktısı hangi ağaç gezinme yöntemine eşdeğerdir? (20P) Yanlış cevaptan 5P kırılacaktır.

- (A) inorder
- (B) preorder
- (C) postorder



3. Düğümleri dairesel bağlı listeye score değerlerine göre küçükten büyüğe sıralı ekleyen insertOrdered() fonksiyonunda ile temsil edilen satır için aşağıda önerilen kodların başına doğru ise **D**; hatalı ise **H** yazınız. (20P) Yanlış cevabın herbirinden 5P kırılacaktır.

(...) back == cursor

(...) back == cursor->next

(...) front == cursor

(...) front == cursor->next