



CEVAPLAR - B

```
void bitOrder(Node* v)
{
    if (v->right != NULL)
    {
        cout << v->elt << " ";
        bitOrder(v->right);
    }
    else
        cout << v->elt << " ";

    if (v->left != NULL)
        bitOrder(v->left);
}

void main()
{
    LinkedListTree 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);

    cout<< "Reverse Preorder Traversal : " ;
    bitOrder(Tree.root);
}
```

1. Yukarıdaki programın çıktısı nedir? (40P)

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

```

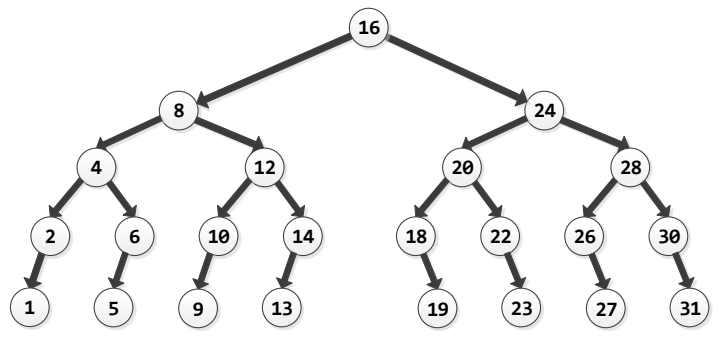
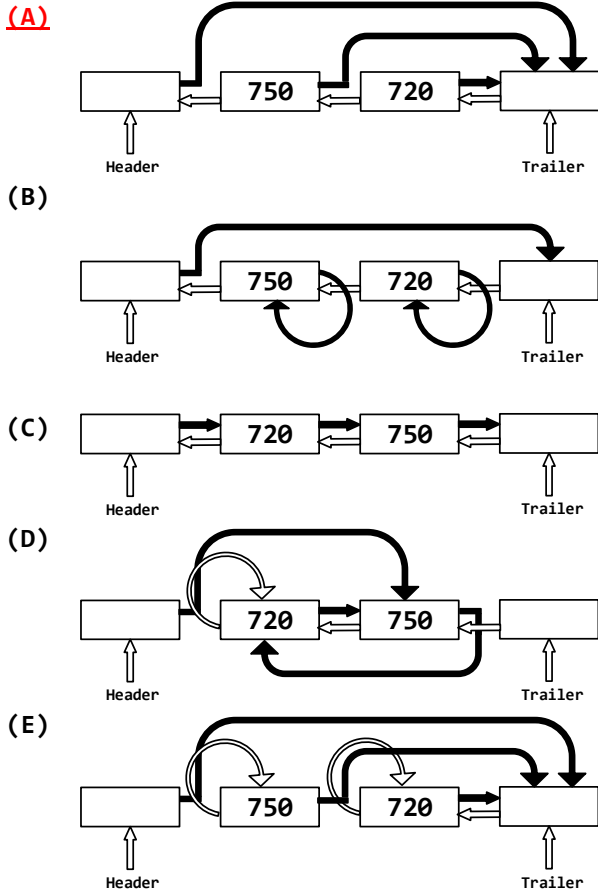
void addFront(const int& i)
{
    add(header->next, i);
}

void add(DoublyNode* v, int& i)
{
    DoublyNode* u = new DoublyNode;
    u->score = i;
    u->prev = v->prev;
    v->prev = u;
    v->prev->next = u;
    u->next = v;
}

void main()
{
    DoublyLinkedList list;
    list.addFront(750);
    list.addFront(720);
}

```

2. addFront() çağrıları sonrası listenin son hali hangisidir?
(Yanlış cevaptan 5P kılacaktır) (30P)



```

if (p->left != NULL) // p silinene işaret eder
{
    temp = p->left;
    while (temp->right != NULL) temp = temp->right;
    p->elt = temp->elt;

    if (temp->left != NULL)
    {
        temp->par->right = temp->left;
        temp->left->par = temp->par;
    }
    else
    {
        temp->par->right = NULL;
    }

    delete temp;
    return;
}

```

3. Yukarıdaki koda göre ağaçtan 8 silindiğinde son hali ne olur? Ağacın tamamını çizin. (30P)

