



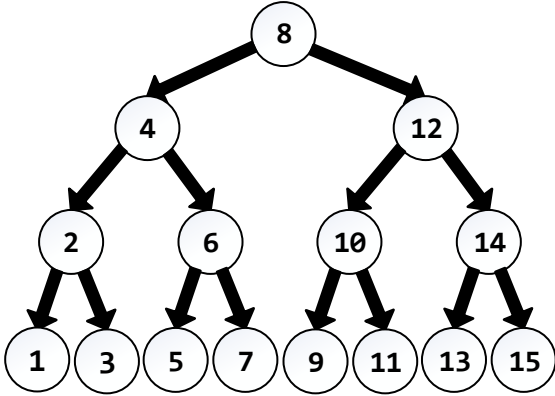
NUMARA : .....	AD SOYAD : .....	DEĞERLENDİRME	
		[.....]	.....
Öğrenciler, <a href="#">Mühendislik Fakültesi Sınav Uygulama Yönergesi</a> 'ndeki kurallara uymalıdır. Sınav Soruları <a href="#">Bölüm Program Çıktıları</a> 'ndan 1,4,12 ile ilişkilidir.			

```
void traverse(TreeNode* v)
{
    if (v->left != NULL)
    {
        traverse(v->left);
        cout << v->elem << " ";
    }

    if (v->right != NULL)
    {
        cout << v->elem << " ";
        traverse(v->right);
    }
}
```

1. main()'de aşağıdaki ağacın rootu ile çağrıldığı varsayılan traverse() fonksiyonunun çıktısı nedir? (40P)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--



8 7 6 5 4 3 2 1

2. Yukarıdaki verilerin ikili ağaca eklendiği varsayılın. Bu ağacın inorder, preorder ve postorder gezinme çıktılarından hangisi diğer ikisinden farklıdır? (30P)  
Yanlış cevaptan 5P kılacaktır.

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

```

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

    DoublyNode* current      = header;

    while (current->next != trailer)
    {
        if (newNode->score >= current->next->score)
            current = current->next;
        else
            break;
    }

    newNode->next             = current->next;
    newNode->prev             = current;
    .....                   = .....;
    .....                   = .....;
}

```

3. insertOrdered() fonksiyonunu tamamlayınız. (30P)

*Yanlış cevaptan 5P kırılacaktır.*

- (A) newNode->next->prev = newNode;  
current->next->prev = newNode;
- (B) newNode->prev->next = newNode;  
current->next = newNode;
- (C) current->next->prev = newNode;  
newNode->prev->next = newNode;
- (D) current->next = newNode;  
current->next->prev = newNode;
- (E) newNode->prev->next = newNode;  
current->next->prev = newNode;