

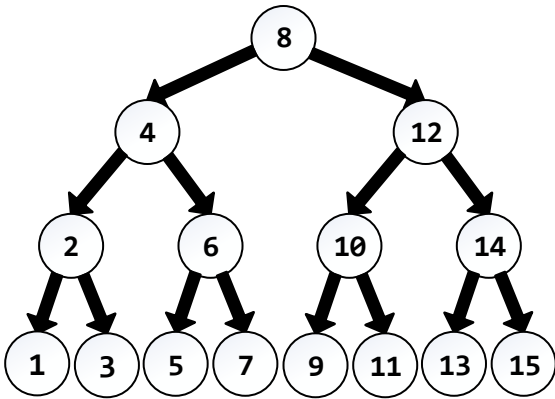


| | | | |
|--|------------------|---------------|-------|
| NUMARA : | AD SOYAD : | DEĞERLENDİRME | |
| | İMZA : | [.....] | |
| Mühendislik Fakültesi Sınav Uygulama Yönergesi'ndeki kurallara uyulmalıdır. Sorular Bölüm Program Çıktıları'ndan 1,4,12 ile ilişkilidir. | | | |

```
void traverse(Node* v)
{
    if (v->left != NULL)
    {
        traverse(v->left);
    }

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

1. main()'de aşağıdaki ağacın rootu ile çağrıldığı varsayılan yukarıdaki traverse() fonksiyonunun çıktısı nedir? (25P)



```
void traverse(Node* v)
{
    stack<Node*> stl_stack;
    Node* current = v;

    while (true)
    {
        if (current != NULL)
        {
            stl_stack.push(current);
            current = current->left;
        }
        else
        {
            if (stl_stack.empty())
            {
                return;
            }
            else
            {
                current = stl_stack.top();

                if ((current->right != NULL)
                    && (current->left != NULL))
                {
                    cout << current->elt << " ";
                }

                stl_stack.pop();
                current = current->right;
            }
        }
    }
}
```

2. main()'de soldaki ağacın rootu ile çağrıldığı varsayılan traverse() fonksiyonunun çıktısı nedir? (25P)

```

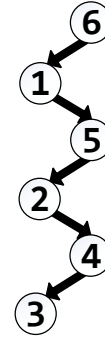
void insertOrdered(DoublyNode* newNode,
                  DoublyNode* current)
{
    if(..... && .....)

        insertOrdered(newNode, current->next);
    else
    {
        newNode->next = current;
        newNode->prev = current->prev;
        current->prev->next = newNode;
        current->prev = newNode;
    }
}

int main()
{
    DoublyLinkedList list; DoublyNode* newNode;
    newNode = new DoublyNode;
    newNode->elem = "Paul"; newNode->score = 720;
    list.insertOrdered(newNode, list.header);
    newNode = new DoublyNode;
    newNode->elem = "Rose"; newNode->score = 590;
    list.insertOrdered(newNode, list.header);
    newNode = new DoublyNode;
    newNode->elem = "Anna"; newNode->score = 660;
    list.insertOrdered(newNode, list.header);
    newNode = new DoublyNode;
    newNode->elem = "Mike"; newNode->score = 1105;
    list.insertOrdered(newNode, list.header);
}

```

| |
|---------|
| Zig |
| Zig-Zig |
| Zig-Zig |
| Zig |
| Zig-Zig |
| Zig-Zig |
| Zig |



4. Yukarıdaki işlemlerle oluşturulan Splay Ağacına verilerin hangi sırada eklendiğini bulunuz. (25P)



3. insertOrdered() fonksiyonundaki satırlarına aşağıdaki kodlardan hangisi yazılmalıdır? (25P)

Not → Header ve Trailer'ın score değerini 0 varsayınız.
Yanlış cevaptan 5P kılacaktır.

- (A) `if ((newNode->score >= current->score) && (current != trailer))`
- (B) `if ((newNode->score >= current->next->score) && (current != trailer))`
- (C) `if ((newNode->score >= current->score) && (current->next != trailer))`
- (D) `if ((newNode->score >= current->next->score) && (current->next != trailer))`