



NUMARA :	AD SOYAD :	DEĞERLENDİRME
	İMZA :	[.....]

Öğrenciler, [Mühendislik Fakültesi Sınav Uygulama Yönergesi](#)'ndeki kurallara uymalıdır.

Sınavın başında sorular kısaca açıklanacaktır. Öğrencilerin soruları cevaplandıktan sonra sınav boyunca soru sormak yasaktır.

```
void traverse1(Node* v)
{
    if (v->left != NULL)        traverse1(v->left);
    else
        if (v->right != NULL)   cout << v->elt << " ";
        traverse1(v->right);
}

void traverse2(Node* v)
{
    if (v->left != NULL)        traverse2(v->left);
    if (v->right != NULL)       traverse2(v->right);
    else
        cout << v->elt << " ";
}

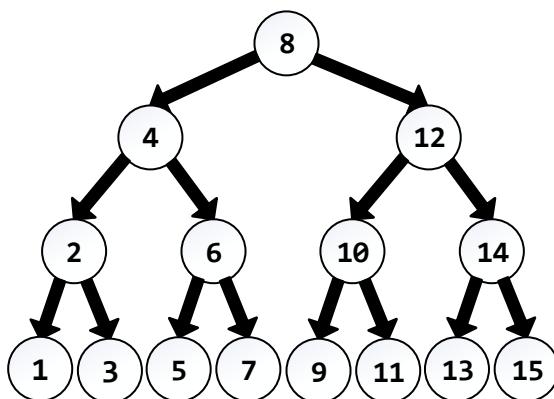
void traverse3(Node* v)
{
    if (v->left == NULL)        cout << v->elt << " ";
    else
        if (v->right != NULL)   traverse3(v->left);
        traverse3(v->right);
}

void traverse4(Node* v)
{
    if (v->left != NULL)        traverse4(v->left);
    if (v->right == NULL)       cout << v->elt << " ";
    else
        traverse4(v->right);
}
```

```
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();
                cout << current->elt << " ";
                stl_stack.pop();
                current = current->right;
            }
        }
    }
}
```

1. `main()`'de aşağıdaki ağacın `root` ile çağrıldığı varsayılan yukarıdaki fonksiyonların çıktıları nelerdir? (20P)



2. `main()`'de soldaki ağacın `root` ile çağrıldığı varsayılan `traverse()` fonksiyonunun çıktışi nedir? (30P)

traverse1:

traverse2:

traverse3:

traverse4:

```

void removeOrdered(const string& e, const int& i)
{
    if(head == NULL)
    {
        cout << "List is empty !" << endl; return;
    }

    if ((strcmp((char*)&head->elem, (char*)&e))
        && (head->score == i))
    {

        SinglyNode* temp = head;
        head = head->next;
        delete temp;
        return;
    }

    SinglyNode* current = head;

    while (current->next != NULL)
    {
        if ((strcmp((char*)&current->next->elem,(char*)&e))
            && (current->next->score == i))
        {

            .....
            .....
            .....

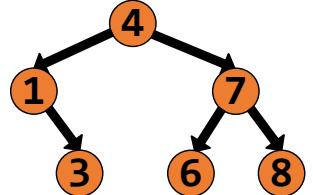
            return;
        }

        current = current->next;
    }

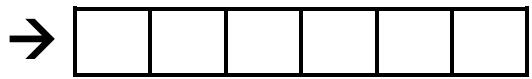
    if (current->next == NULL)
        cout << "\n" << e << " is not found" << endl;
}

```

Zig	(X:Sol)
Zig-Zig	(X:Sağ, P:Sağ)
Zig-Zig	(X:Sol, P:Sol)
Zig	(X:Sol)
Zig-Zig	(X:Sağ, P:Sağ)
Zig-Zag	(X:Sol, P:Sağ)
Zig-Zag	(X:Sağ, P:Sol)
Zig-Zag	(X:Sağ, P:Sol)



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



3. removeOrdered() fonksiyonunu tamamlayınız. (25P)

İpucu → Silinecek eleman **current->next** 'tir.