



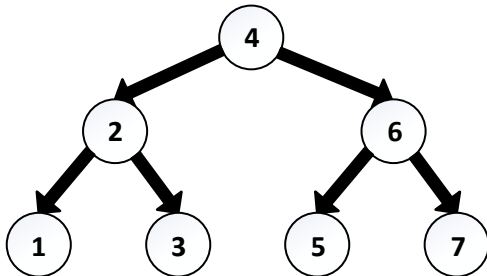
NUMARA : .....		AD SOYAD : .....		DEĞERLENDİRME	
Sınavda Uyulması Gereken Kurallar		İMZA : .....		[.....]	.....
1. Cep telefonlarının saate bakmak için bile olsa herhangi bir amaçla kullanılması yasaktır. Telefon kapalı ve cepte olmalıdır. 2. Sınavın başında sorular kısaca açıklanacaktır. Öğrencilerin soruları cevaplandıktan sonra sınav boyunca soru sormak yasaktır. 3. Soru kağıdına numaranızı ve isminizi yazıp imzalamayı unutmayınız.					

```
void traverse(Node* v)
{
    cout << v->elt << " ";
    if (v->left != NULL)
    {
        traverse(v->left);           // A
        cout << v->elt << " ";       // A
    }
    if (v->right != NULL)
    {
        traverse(v->right);          // B
        cout << v->elt << " ";       // B
    }
}
```

1. traverse() fonksiyonunun main()'de aşağıdaki ağacın rootu ile çağırıldığı varsayıldığında:

a) // A ile temsil edilen kıvrık parantezler kapatılırsa çıktı ne olur? (15P)

b) // B ile temsil edilen kıvrık parantezler kapatılırsa çıktı ne olur? (15P)



```
void addBack(const string& e, const int& i)
{
    CircularlyNode* v = new CircularlyNode;
    v->elem = e;
    v->score = i;

    if (cursor == NULL)
    {
        v->next = v;
        cursor = v;
    }
    else
    {
        ..... = ..... ;
        ..... = ..... ;
        ..... = ..... ;
    }
}
```

2. Dairesel listenin sonuna düğüm ekleyen addBack() fonksiyonunu tamamlayınız. (20P)

```

void insertOrdered(const string& e, const 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      = .....;
    newNode->prev      = .....;
    ..... = newNode;
    ..... = newNode;
}

int main()
{
    DoublyLinkedList list;

    list.insertOrdered("Paul", 720);
    list.insertOrdered("Rose", 590);
    list.insertOrdered("Anna", 660);
    list.insertOrdered("Mike", 1105);
    list.insertOrdered("Rob" , 750);
    list.insertOrdered("Jack", 510);
    list.insertOrdered("Jill", 740);
}

```

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

```

#include <iostream>
#include <stack>
using namespace std;

void main()
{
    stack<int> stl_stack;

    int temp = 61;

    while (temp != 0)
    {
        stl_stack.push(temp % 2);
        temp = temp / 2;
    }

    while (!stl_stack.empty())
    {
        if (stl_stack.top() == 1)
            cout << '1';
        else
            cout << '0';

        stl_stack.pop();
    }

    getchar();
}

```

4. Yukarıdaki programın çıktısı nedir? (20P)