

Worksheet 4

Find His Dairy (V1.0)

Deskripsi

Yuno dan Amano adalah sepasang kekasih yang romantis, harmonis, saling melengkapi, dan rajin menabung. Keduanya memiliki hobi yang sama yaitu menulis catatan harian atau diary. Hubungan kedekatan mereka sudah terdengar kemana-mana bahkan hingga ke pelosok yuli. Namun, suatu hari sikap Amano terhadap Yuno berubah, dari yang awalnya perhatian dan sering memberikan kabar menjadi sulit dihubungi dan menghindar untuk bertemu dengan Yuno. Yuno merasa ada suatu hal yang terjadi pada Amano sehingga sifat Amano berubah. Yuno mengetahui bahwa Amano suka menulis catatan harian(diary) pada smartphone-nya. Yuno juga mengetahui bahwa Amano menyimpan diary tersebut di dalam folder/direktori yang bernama Diary. Akhirnya Yuno memutuskan untuk mencari tahu isi diary Amano.

Melalui sebuah program yang didapatkannya dari internet, Yuno berhasil mendapatkan **dua buah daftar folder-folder** yang terdapat pada smartphone Amano. Akan tetapi kedua daftar folder/direktori tersebut tersusun secara random sehingga Yuno tidak dapat mengetahui lokasi pasti folder **Diary** yang dicarinya. Pada akhirnya Yuno meminta bantuan temannya, yaitu Anda, untuk membantunya **menemukan lokasi folder Dairy** jika diberikan informasi daftar folder-folder dan **setiap folder maksimal memiliki dua buah subfolder**. Sebagai teman yang baik Anda bersedia untuk membantu Yuno dengan senang hati.

Setelah melakukan riset selama 3 hari 3 malam, Anda tidak bisa menemukan lokasi folder Diary karena daftar folder yang diberikan tersusun secara acak. Akhirnya Anda memutuskan untuk tidur dan beristirahat. Namun, ketika Anda sedang tidur, di dalam mimpi Anda diberitahu oleh seseorang bahwa daftar folder tersebut tidak tersusun secara acak. Namun **kedua daftar folder tersebut tersusun secara postorder dan inorder**. Dari mimpi tersebut, anda menyadari bahwa melalui data postorder dan inorder dapat disusun sebuah binary tree. Kemudian dari binary tree tersebut, Anda dapat menemukan lokasi folder Diary.

Format Masukan

Baris pertama berisi sebuah bilangan bulat N yaitu banyak data nama folder yang diberikan. Baris kedua berisi N buah nama folder yang telah terurut secara postorder. Baris ketiga berisi N buah nama folder yang telah terurut secara inorder yang telah di jelaskan pada bagian deskripsi.

Format Keluaran

Urutan folder yang harus dibuka Yuno agar sampai ke folder Diary

Batasan

- Nama folder yang diberikan hanya terdiri dari satu kata.
- Nama folder dijamin terdiri dari huruf alfabet (a-z atau A-Z) saja tanpa simbol apapun
- Dijamin tidak ada nama folder yang sama (perlu diperhatikan bahwa nama folder case sensitive)
- Pasti terdapat folder Diary pada input

Contoh 1

Masukan

7

Diary folderD folderB folderE folderF folderC folderA

Diary folderB folderD folderA folderE folderC folderF

Keluaran

folderA

folderB

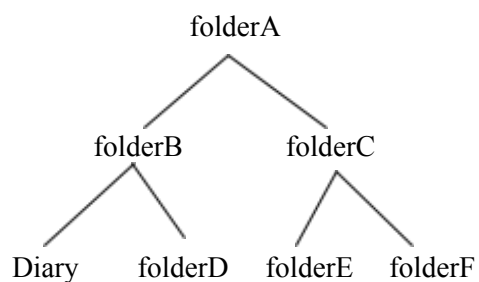
Diary

Penjelasan

Urutan postorder : Diary folderD folderB folderE folderF folderC folderA

Urutan inorder : Diary folderB folderD folderA folderE folderC folderF

Binary Tree yang dibentuk dari urutan postorder dan inorder diatas adalah:



Contoh 2

Masukan

14

F D H K L J I Diary G E B M C A

F D B H G K J L I Diary E A M C

Keluaran

A

B

E

G

Diary

Template Code

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.StringTokenizer;

public class WS4 {

    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(reader.readLine());
        String postorder = reader.readLine();
        String inorder = reader.readLine();

        StringTokenizer stPostorder = new StringTokenizer(postorder);
        StringTokenizer stInorder = new StringTokenizer(inorder);

        BinaryTree tree = new BinaryTree();
        //@TODO : Do something

        tree.findDiary();
    }
}

class Node {
    String data;
    Node left, right, parent;

    public Node(String data){
        this.data = data;
        left = right = parent = null;
    }
}

class BinaryTree {
    // Tambahkan attribut yang kalian butuhkan
    static int index;

    Node buildTreeFromPostIn(ArrayList<String> in, ArrayList<String> post, int start,int end, Node
parent) {
        //@TODO : Implementasi pembuatan binary tree dari array inorder dan postorder
        // Silahkan tambahkan/hilangkan paramater
        return ;
    }
}
```

```
// menginisialisasi pembuatan binary tree
Node buildTree(ArrayList<String> in, ArrayList<String> post, int n){
    index = n - 1;
    return buildTreeFromPostIn(in, post, 0, n - 1, null);
}

// Mencari folder diary
Node find(String namaFolder){
    //@TODO: implementasi proses pencarian folder Diary. Pencarian bisa dilakukan dengan
    traversal atau cara lainnya
    return;
}

// Mencetak folder apa saja yang harus dibuka
void findDiary(){
    Node dairy = find("Diary");
    //@TODO: lengkapi method untuk menghasilkan outpus sesuai soal
}
}
```