

Below is our coding challenge to keep things moving forward.  Take your time to complete this one – it’s not timed and we aren’t looking for you to crack the code quickly, it’s more about your best response please.

**Question 1 - Finding the best plan:**

             Let’s say a cloud service offers different plans. For example, Office 365 has [different plans](http://www.microsoft.com/en-us/office365/compare-plans.aspx) called E1, F1, etc. Each plan has different features such as voice, email, archiving, etc. Now let say the user selects a set of features he/she wants. The goal is to write code in Java that finds the combination of plans that offers all selected features at the lowest price. Note that in some cases, it will be just one plan, but in other cases you will need 2 or more plans to get all the features you want.

To get you started:

class Feature { public string Name; }

              class Plan { public string Name; public double Cost; public Feature[] Features; }

             Plan[] allPlans;            // this is the list of plans available instantiated as per the above

Feature[] selectedFeatures;       // this is the list of features the user wants -> find combinations of 1-N plans that fulfil those features -> select the cheapest combination(s).

**Question 2 - Linking nodes in a tree:**

             Each Node represents an element of a tree and specifies a list of immediate children. The 'Children' property lists all children (in order) but the 'Right' property is set to null. Suppose you are given the root of a fully populated tree (i.e. a Node instance called rootNode). Write code in Java to set the 'Right' property so that each node is linked to right siblings without using a queue or stack (do not use recursive calls). Make sure to test your code with the sample tree below.

To get you started:

public class Node

{

                public Node[] Children;

                public Node Right;

}

Node rootNode;

