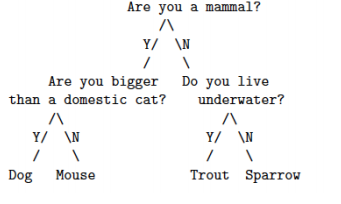
#### Problem Definition

The goal of the program is to implement one of the most famous game “Animal Guess”. Here, a player thinks of an animal; the program tries to guess the animal depending on the responses the player provides. e.g.: The program asks the player whether the animal is larger than a cat? If the player provides a response, the program goes on diving further deep into the tree node. When the program reaches the leaf node, there would have been enough evidence to guess the animal correctly.



#### Methodology

#### Data Structure

The data structure preferred is a linked list. The nodes of linked list have been implemented as instances of Node Class.

Public class Node {

Node leftChild;

Node rightChild;

String value;

Integer id;

}

Here,

leftChild → represents left Child of the node

rightChild → represents right Child of the node

value → represents value inside the node, may be a question

id → id that represents the node, easy to identify

#### Algorithm

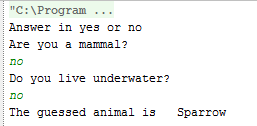
1. Start
2. Store all the data to be input into the linked list in such a way that you are following the trick mentioned above.
3. Add the data of the array as nodes of the linked list following a recursive function call technique. That is, starting by adding the middle index array value as node of the linked list, then calling the recursive function for elements to the left of the middle index and doing the same for the elements to the right of the middle index, continuing till start index! = end index.
4. After linked list is formed, the game is started. Until the leaf node is reached, the questions (data in nodes) are asked and depending upon the response (yes/no), traversal is done to the left or right sub-tree.
5. Final response is given based on which leaf node is reached through traversal.
6. End

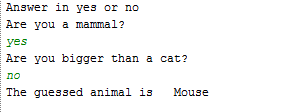
#### Annex

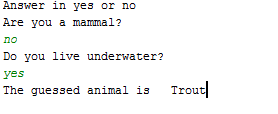
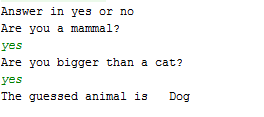
The implementation of this problem using Java is given below:

*/\*\*  
 \* Created by mohan on 4/25/2016.  
 \*/***import** java.util.Scanner;  
**public class** AnimalGuessing {  
  
 *// Node is a class that contains information about a node.* **public class** Node {  
 Node **leftChild**,**rightChild**;  
 Integer **id**;  
 String **question**;  
 **public** Node(String question, **int** id) {  
 **this**.**question** = question;  
 **this**.**id** = id;  
 }  
 }  
  
 *//This function is used to add root Node* **public** Node addRootNode(String question,**int** id)  
 {  
 Node rootNode=**new** Node(question,id);  
 **return** rootNode;  
 }  
  
 *//This functions creates and add the left child to the node for the given id* **public void** addLeftNode(Node rootNode, **int** parentId, **int** id, String question) {  
 **if** (rootNode.**id** == parentId) {  
 Node newNode = **new** Node(question, id);  
 rootNode.**leftChild** = newNode;  
 **return**;  
 }  
 **else** {  
 **if** (rootNode.**leftChild** != **null**) {  
 addLeftNode(rootNode.**leftChild**, parentId, id, question);  
 }  
 **if** (rootNode.**rightChild** != **null**) {  
 addLeftNode(rootNode.**rightChild**, parentId, id, question);  
 }  
 }  
 }  
  
 *//This functions creates and add the left child to the node for the given id* **public void** addRightNode(Node rootNode, **int** parentId, **int** id, String question) {  
 **if** (rootNode.**id** == parentId) {  
 rootNode.**rightChild** = **new** Node(question, id);  
 **return**;  
 } **else** {  
 **if** (rootNode.**leftChild** != **null**) {  
 addRightNode(rootNode.**leftChild**, parentId, id, question);  
 }  
 **if** (rootNode.**rightChild** != **null**)  
 addRightNode(rootNode.**rightChild**, parentId, id, question);  
 }  
 }  
  
 **public static void** main(String[] args) {  
  
 AnimalGuessing addNode = **new** AnimalGuessing();  
 *//Root node is created as* Node rootNode=addNode.addRootNode(**"Are you a mammal?"**,0);  
  
 *//The nodes are added to the tree as* addNode.addLeftNode(rootNode, 0, 1, **"Are you bigger than a cat?"**);  
 addNode.addLeftNode(rootNode, 1, 3, **"Dog"**);  
 addNode.addRightNode(rootNode, 1, 4, **"Mouse"**);  
  
 addNode.addRightNode(rootNode, 0, 2, **"Do you live underwater?"**);  
 addNode.addLeftNode(rootNode, 2, 5, **"Trout"**);  
 addNode.addRightNode(rootNode, 2, 6, **"Sparrow"**);  
  
 System.***out***.println(**"Answer in yes or no"**);  
 *//rootnode is passed to startGame  
 startGame*(rootNode);  
 }  
  
 *//This function check if the given node is leaf node or not* **public static boolean** checkLeaf(Node node)  
 {  
 **boolean** isLeaf=**false**;  
 **if**(node.**leftChild**==**null** && node.**rightChild**==**null**)  
 {  
 isLeaf=**true**;  
 }  
 **return** isLeaf;  
 }  
  
 **public static void** startGame(Node node) {  
 **while** (**true**) {  
  
 Scanner input = **new** Scanner(System.***in***);  
 System.***out***.println(node.**question**);  
 String answer = input.nextLine().toLowerCase();  
 **if** (answer.equals(**"yes"**)) {  
 node=node.**leftChild**;  
 } **else if** (answer.equals(**"no"**)) {  
 node=node.**rightChild**;  
 } **else** {  
 System.***out***.println(**"Invalid entry! Enter either yes or no only."**);  
 }  
 **if**(*checkLeaf*(node) == **true**)  
 {  
 System.***out***.println(**"The guessed animal is \t"**+node.**question**);  
 **break**;  
 }  
 }  
 }  
}

# Output





#### Analysis

The question is asked starting with the question in the root node (i.e. “Are you a mammal?”). Then according to the response (yes or no) other questions are asked. If the response is yes left sub tree is traversed else right sub tree is traversed. After reaching to the leaf node we stop and the animal is found.