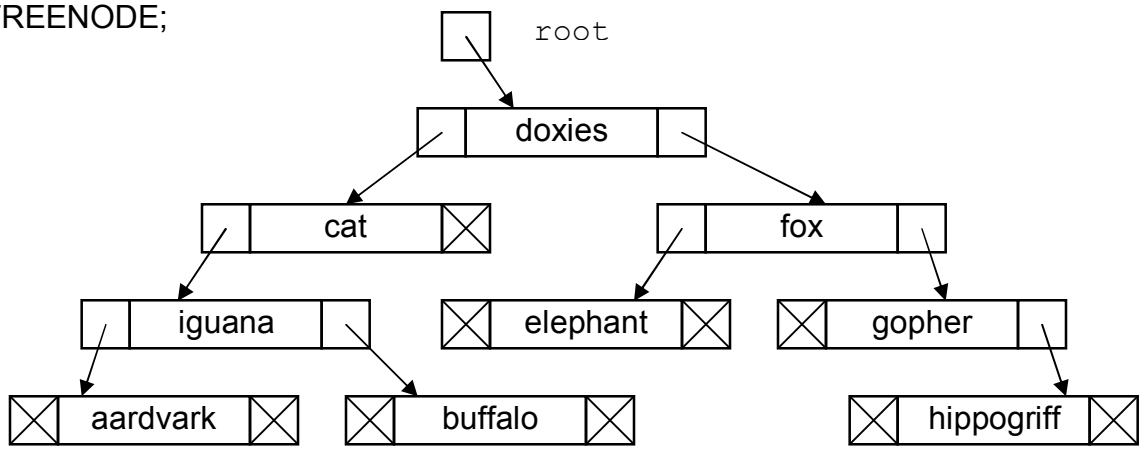
University of Massachusetts Dartmouth
Department of Electrical and Computer Engineering

ECE 161 Name: extratree.cpp
Extra Credit Due: Not later than May 3, 2013 11:59pm

In the previous problem, we looked at a tree structure. The sample tree in the previous handout is repeated below. The definition of TREENODE is also presented.

```
typedef struct treenode
{
        struct treenode *left;
        char name[30];
        struct treenode *right;
} TREENODE;
```



A first step of the extra credit is to rewrite the printtree() function such that it prints a "sideways" representation of the tree. If the example above were the tree, then printed should be:

```
            hippogriff
        gopher
    fox
        elephant
doxies
    cat
            buffalo
        iguana
            aardvark
```

The program must read from a data file named `extratree.txt`. The first line of the file is the total number of nodes (`N`) in the tree. For this example, the number of nodes would be 9. The remainder of the file shall be `N*2` lines (total length of file shall be `N*2+1` lines. The next `N` lines shall be the listing if the tree was printed out with a simple pre-fix order. The next `N` lines after that shall be the listing if the file was printed with a simple in-fix order. Simple pre and in-fix print routines are provided on the next page for reference.

Sample:

If the file `extratree.txt` contains:

```
9
doxies
cat
iguana
aardvark
buffalo
fox
elephant
gopher
hippogriff
aardvark
iguana
buffalo
cat
doxies
elephant
fox
gopher
hippogriff
```

Then the output of the program should be:

```
            hippogriff
        gopher
    fox
        elephant
doxies
    cat
            buffalo
        iguana
            aardvark
```

Notes:

```
// Pre-fix print routine:
void PrintTree(TREENODE *r)
{
     if (r)
     {
          printf("%s\n",r->name);
          PrintTree(r->left);
          PrintTree(r->right);
     }
}
```

```
// in-fix print routine:
void PrintTree(TREENODE *r)
{
     if (r)
     {
          PrintTree(r->left);
          printf("%s\n",r->name);
          PrintTree(r->right,);
     }
}
```