C/C++ 2014/15 programming exercise 1

Part 1: easy

Here is a struct that can represent binary trees with data at the nodes:

```
struct twoptr {
    struct twoptr *left;
    int data;
    struct twoptr *right;
};
```

Write a C function

```
void toxml(struct twoptr *p)
```

that prints binary trees, as represented with the struct above, in the form of properly indented XML. Null pointers are not represented in the output.

For example, suppose we have a binary tree with three nodes, such that the top one has 2 as its data member and its two children have 5 and 7 as their data members. This tree should be printed as follows:

```
<twoptr>
    <twoptr>
        5
    </twoptr>
    2
    <twoptr>
        7
    </twoptr>
</twoptr>
```

Hint: it is straightforward to write a recursive function that takes the current level of indentation as a parameter. [7 points]

Part 2: a little harder

There are situations, like memory management, where one needs to traverse pointers without incurring the (small but in this case significant) overhead of recursive function calls. For the same kinds of trees as above, write a C function

```
void traverse(struct twoptr *curr)
```

that prints the data at the nodes of a given tree, but **without using recursion**. You are not required to print any XML tags here. So for the above example, the output should be:

```
5
2
7
```

Hint: use a while-loop and a stack of pointers instead of recursive calls. To understand what is going on, it helps a lot if you draw a diagram of the tree together with your stack data structure. [3 points]