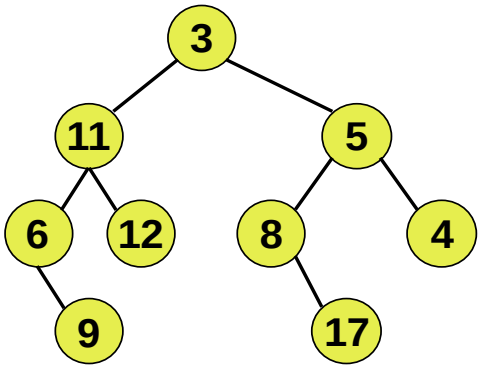
The Tasks

- In this assignment, you need to implement a binary tree class (class name: **`BT`**) using the left-child-right-child representation. You do not need to make a class template; simply assume that the **`data`** field of a tree node is of type **`int`**.
- Specifically, there are two main tasks that you need to do:
  - Initialization of the tree from a text string. Details are explained in next page.
  - By adding a **`parent`** field in each node, implement a function for inorder traversal using no recursion, no stack, and no threads.

The Tasks (Tree Initialization)

- Implement a class constructor that takes a constant text string for its input. The string should contain a series of integers separated by blanks. The first integer is the height of the tree. The remaining integers are the values of the tree nodes' data fields. These integers are listed in the order as if they are in a full binary tree. A negative number in the list indicates that the place has no node.
- Example tree:



- The input string for the example tree:

```
4 3 11 5 6 12 8 4 -1 9 -1 -1 -1 17
```

The Tasks (Inorder Traversal)

- Each node should have a **`parent`** pointer, which points to its parent. (It's **`NULL`** for the root.) Be sure to set these pointers correctly when you initialize the tree.
- Implement a member function **`inorder`** that does inorder traversal of the tree. Just print out a node's **`data`** field (an integer) when "visiting" that node.
- You should implement this function without the use of recursion, stacks, or threads.
- Hint: You can think about how you can implement the functionality of **`InorderSuccessor`** using the **`parent`** field, and put this function in a while loop for inorder traversal.

The Guidelines

- Allowed environments: VS2012/2013/2015, Dev-C++. Indicate your environment at the beginning of your code.
- You need to write your own **`main`** function to test your permutation generation function. You do not need to include this **`main`** function in your submission. The instructor will provide a test **`main`** function for you.
- No usage of STL class templates allowed.
- Include documentation; this will be part of your grade.
- Demo: Only a randomly selected subset of students; will be announced separately after the due date.

The Guidelines

- Submission:
  - Use E3 only.
  - Submit all your code in a single header file (`.h`). Name it `P3_xxxxxx.h`, where `xxxxxx` is your ID. **Do not** submit your `main` function or any file that is not your code (such as the *.sln file). No compressed file (*.zip, *.rar, etc.). **Only the header file!!!**
  - Due date: **12/4/2015**. There's a grace period of 4 days with 10% deduction per day. (The deduction kicks in only when you have accumulated more than three days of delay during the semester.)