* 1. Use a top-down approach to build a heap from the following array. **Show the intermediate results after adding each element to the heap (You should have 15 pictures).** Show how the heap will be stored in the array at the end.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 17 | 1 | 50 | 14 | 65 | 19 | 90 | 7 | 10 | 2 | 12 | 4 | 70 | 40 |

* 1. Using the resulting heap and it is corresponding array representation from part *a*, show how the array would look like after the first, second and third iteration of applying heap-sort to the array.

1. Given the following definition of node for a binary tree:

//Definition of the node

struct nodeType

{

int info;

nodeType \*llink;

nodeType \*rlink;

};

Using the following definition for the function visit(), show how the following binary tree will look like after applying the inorder() traversal on the following tree. What is the output of the in-order traversal?

visit(nodeType \*p)

{

if ((p->llink!= NULL) && (p->info < 20))

{

p->info + = 2;

cout(p->info); // print the info the node

}

}

