```
2
3
Output2:
3 5 7 9 11
The queue is full.
7 9 11
```

9. Implement a binary search tree with linked lists

Please implement a binary search tree with linked lists and traversal with in-order, post-order, pre-order. The input consists of a sequence of numbers. You must build a binary search tree according to the input order. The output includes three lines of the results: (1) the in-order traversal, (2) the post-order traversal, and (3) the pre-order traversal.

Note: You must use linked lists; otherwise, no points will be given.

Test Case

Please test your program with Input, and then check the answers with Output.

Listing 9 : Implement a binary search tree with linked list

```
Input：
4 6 8 9 12 5 7

Output：
4 5 6 7 8 9 12
5 7 12 9 8 6 4
4 6 5 8 7 9 12
```

10. Insert and delete a number of binary search tree

Implement a binary search tree with insert(x) and delete(x) functions using an array. You must build a binary search tree according to the input order. The input consists of a sequence of numbers in the first line. The insert/delete function and the number to be insert/deleted are shown in the remaining lines. For the delete operation, you must use the node with the largest key from the left sub-tree to replace the deleted node. Finally, print out the entire tree in level order from the top to the down. If there are more than two numbers in the same level, you should print out the numbers from the left to the right

Test Case

Please test your program with Input, and then check the answers with Output.

Listing 10 : Insert and delete a number of binary search tree

```
Input：
4 5 1 0 6 7 2
insert(44)
delete(6)
insert(3)
delete(2)

Output：
4 1 5 0 3 7 44
```

11. Heap sort

Given a sequence of n numbers, build a heap using an array and implement a heap sort that results in a sorted numbers in ascending order. The output includes two lines of the results: (1) the contents of the array that stores the heap in each step, and (2) the sorted numbers in ascending order.

Test Case

Please test your program with Input1 and Input2, and then check the answers with Output1 and Output2.

Listing 11: Heap sort

```
Input1：
1 4 5 0 6 7 2

Output1：
0 1 2 4 6 7 5
1 4 2 5 6 7
2 4 7 5 6
4 5 7 6
5 6 7
6 7
7
0 1 2 4 5 6 7
```