Ex1: Consider the following pseudo code. Assume that IntQueue is an integer queue. What does the function fun do?

```
void fun(int n)
{
    IntQueue q = new IntQueue();
    q.enqueue(0);
    q.enqueue(1);
    for (int i = 0; i < n; i++)
    {
        int a = q.dequeue();
        int b = q.dequeue();
        q.enqueue(b);
        q.enqueue(a + b);
        ptint(a);
    }
}
```

Soln : Prints first n Fibonacci numbers

Ex2 :

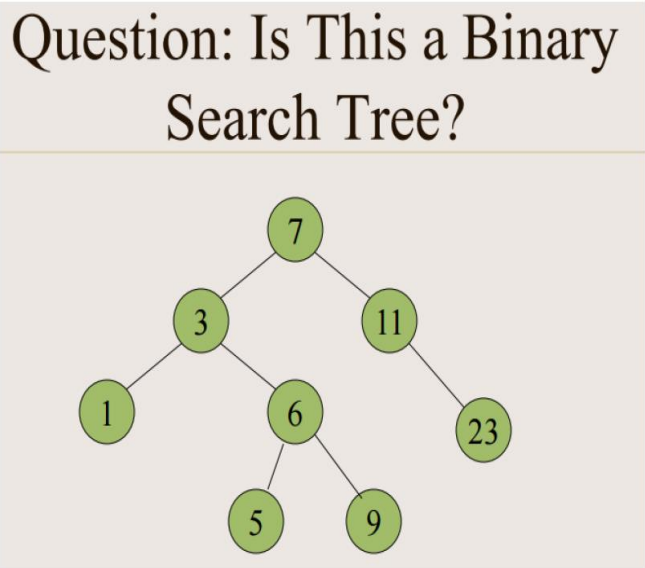
Following is C like pseudo code of a function that takes a Queue as an argument, and uses a stack S to do processing.

```
void fun(Queue *Q)
{
    Stack S;  // Say it creates an empty stack S

    // Run while Q is not empty
    while (!isEmpty(Q))
    {
        // deQueue an item from Q and push the dequeued item to S
        push(&S, deQueue(Q));
    }

    // Run while Stack S is not empty
    while (!isEmpty(&S))
    {
      // Pop an item from S and enqueue the poppped item to Q
      enQueue(Q, pop(&S));
    }
}
```

What does the above function do in general?

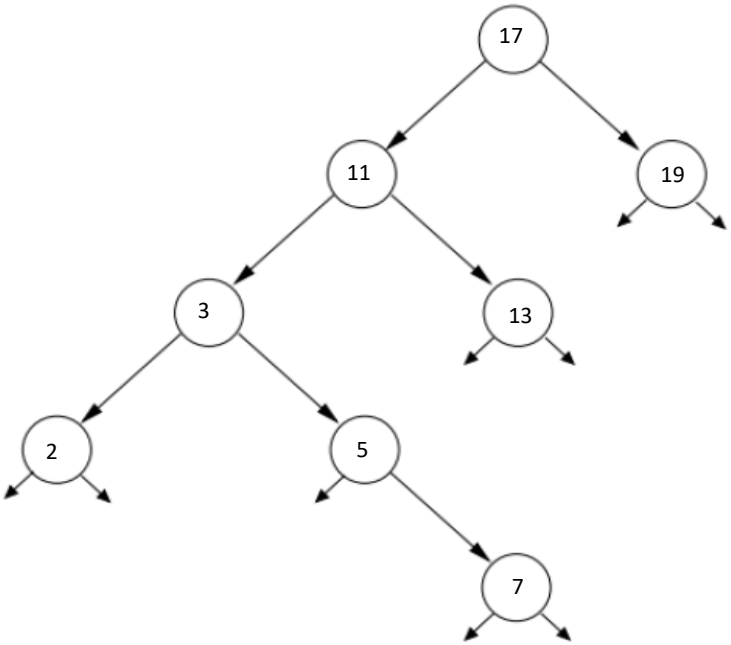
Soln : Reverses the Q

Q1)



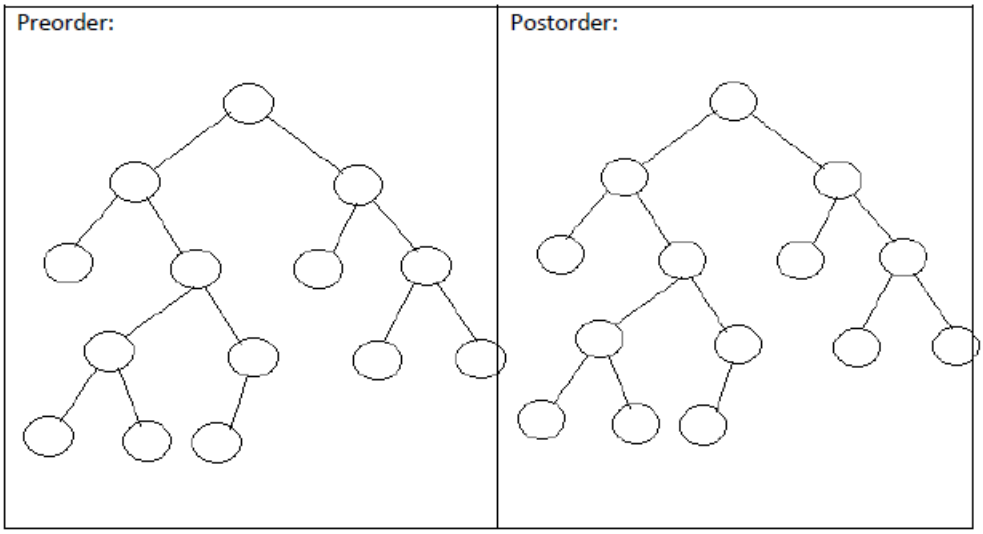
Q2)

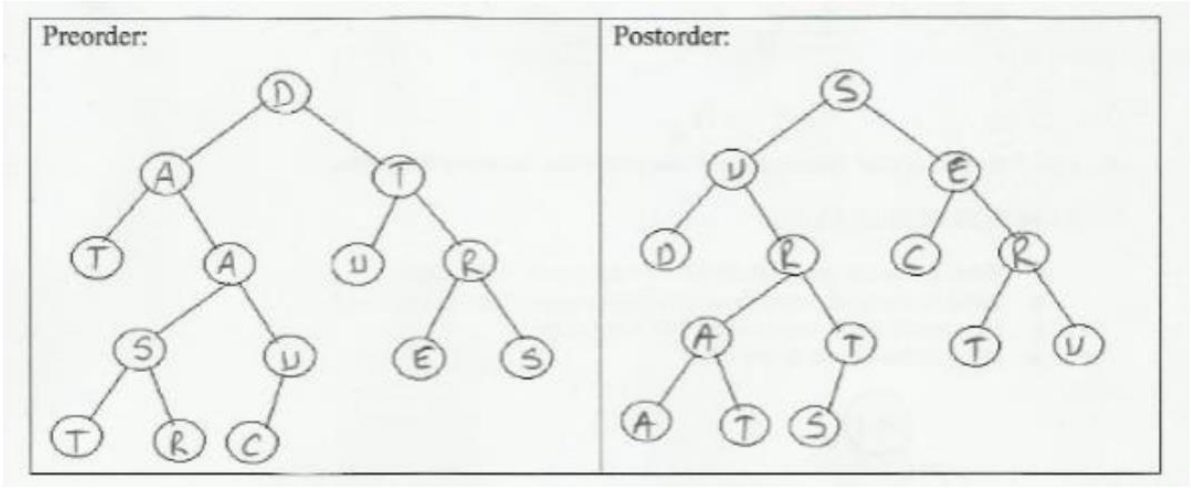
Assign the keys 2, 3, 5, 7, 11, 13, 17, 19 to the nodes of the binary search tree below so that they satisfy the binary-search-tree property.



Q3)

Fill the following trees with "DATASTRUCTURES" statement. The one on the left side must be filled with respect to preorder traversal rule; the one on the right side must be filled with respect to postorder traversal rule.





References

[1] https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-introduction-to-algorithms-sma-5503-fall-2005/exams/final_sol.pdf

[2] http://cslibrary.stanford.edu/110/BinaryTrees.pdf (check this reference for code examples)

[3] http://academic.regis.edu/dbahr/generalpages/datastructures/datastrucpart10.pdf

[4] http://www.cs.toronto.edu/~hojjat/148s07/tests/PastExams/20051ans.pdf

[5] https://www.csee.umbc.edu/courses/undergraduate/341/fall12/exams/Binary-Search-Tree-Review.pdf