ESO207 Assignment 2
Submit on Sunday 25/06/2017. More details about submission will be posted soon.

Question 1. [Marks 2.5% of the entire course weightage].

In this programming assignment you will implement Red-Black tree data structure. This will include Insert routive, re-balance-routine, pre-fix-Tour routine and Visit routine.

The tree should be implemented using nodes having fields: $data$, $colour$, $parent$, $leftChild$, $rightChild$. Pointer $Root$ points to the root node.

Input: $n, a_1, a_2, \ldots, a_n$, where a_i are integers (both positive and negative).

Procedure description:

1. The procedure reads non-negative integer n and subsequently reads in $a_1, dots, a_n$.
2. Then initializes an empty Red-Black tree.
3. It then inserts/re-balances the integers one by one from a_1 to a_n.
4. Finally it performs the Pre-fix-Tour as follows:

if $root = null$ **then**
  Return "empty tree"
else
  $Visit(Root)$;
end

Algorithm 1: Pre-fix-Tour

if $x \neq null$ **then**
  $Visit(x.leftChild)$;
  $Print(x.data, x.colour, x.parent.data)$;
  `/* print each 3-tuple in a new line. */`
  $Visit(x.rightChild)$;
end

Algorithm 2: $Visit(x)$