Daily Practice ….date 2/11/2023

Problem – Add digits

```
package ishwarchavan.com;

public class Add_digit {     //class created

      public static void main(String[] args)    //main program started
      {
          int n = 38;      //variable initialize and declaration

          System.out.println(addDigits(n));      //calling function
      }
    public static int addDigits(int num) {      //function method created

       return num < 10 ? num : addDigits((num % 10) + ( num /10));    //condition
operator checking

    }
}
```

Problem – Binary Tree Paths

```
package ishwarchavan.com;

class Node {      //A binary tree node has data, pointer to left child and a pointer to
right child
      int data;
      Node left, right;

      Node(int item)
      {
            data = item;
            left = right = null;
      }
}

 public class BinaryTreePaths {      //method created
      Node root;

      void printPaths(Node node) {    //printPaths function created
            int path[] = new int[1000];     //object created
            printPathsRecur(node, path, 0);   //calling function
      }

      void printPathsRecur(Node node, int path[], int pathLen) {    //function created
            if (node == null)     //condition checking  if true then return
                  return;


            path[pathLen] = node.data;     // append this node to the path array
            pathLen++;

            if (node.left == null && node.right == null)     //if true then call
printArray function
                  printArray(path, pathLen);
            else                          //otherwise execute both subtree
            {
                  printPathsRecur(node.left, path, pathLen);
                  printPathsRecur(node.right, path, pathLen);
            }
      }

      void printArray(int ints[], int len) {    //print array function created
            int i;
            for (i = 0; i < len; i++)
            {
                  System.out.print(ints[i] + " ");
            }
            System.out.println("");
      }
      public static void main(String args[]) {    //main program started
            BinaryTreePaths tree = new BinaryTreePaths();    //object created
            tree.root = new Node(10);
            tree.root.left = new Node(8);
            tree.root.right = new Node(2);
            tree.root.left.left = new Node(3);
            tree.root.right.left = new Node(2);

            tree.printPaths(tree.root);
      }
}
```