Daily Practice Problem   Date 16/11/2023

Problem 1: Generate Parentheses

```
package ishwarchavan.com;

public class ParenthesesGen {       //class created

      static void printParenthesis(char str[], int pos, int n, int open, int close)
//function created to print parentheses
      {
            if (close == n) {

                  for (int i = 0; i < str.length; i++)      // print the possible
combinations
                        System.out.print(str[i]);
                  System.out.println();
                  return;
            }
            else {
                  if (open > close) {       //condition checking
                        str[pos] = '}';
                        printParenthesis(str, pos + 1, n, open, close + 1);
                  }
                  if (open < n) {
                        str[pos] = '{';
                        printParenthesis(str, pos + 1, n, open + 1, close);
                  }
            }
      }
      static void printParenthesis(char str[], int n)    // Wrapper over
_printParenthesis()
      {
            if (n > 0)
                  printParenthesis(str, 0, n, 0, 0);
            return;
      }

      public static void main(String[] args)      //main program started
      {
            int n = 2;
            char[] str = new char[2 * n];
            printParenthesis(str, n);      //calling function
      }
}
```

Problem 2: Single Number

```
package ishwarchavan.com;

public class SingleNum  {      //class created
      public static void main(String[] args) {     //main program started
            int [] nums =  {2,2,1};
            int x = singleNumber(nums);    //calling function
            System.out.println(x);
                        }
    public static int singleNumber(int[] nums) {     //function created
        int result=0;
        for(int i=0; i<nums.length; i++) {      //loop iteration
            result = result^nums[i];
        }
        return result;      //return result
    }
}
```