Java Assessment

Problem 1

Given a number between 1 and 2e128 print the number as the sum of the prime numbers using the least repeating possible combination.

Example :

Input : 8
Output: 7 + 1

Input : 15
Output: 7 + 5 + 3

Input : 24
Output : 21 + 3

Input : 11
Output : 7 + 3 + 1
Hint (7 + 2 + 2) is in correct because 2 is repeated and a solution exists without repeating

```
public class PrimeFactors {

    public static void main(String[] args) {
        computePrimeFactors(24);
    }

    public static void computePrimeFactors(int x) {
         //TODO Add your implementation here
         return;
    }
}
```

Problem 2

Given an array, sort the array such that the sum of the numbers of the first half is closer to the sum of the numbers in the 2nd half of the array. The input array is not sorted

Example :

Input : [1, 2, 3, 4]
Output : [1, 4, 3, 2]
Hint: Array has an length of 4 sum of first half is 5 and sum of 2nd half in 5

Input : [2, 7, 3, 8, 11, 12]
Output : [11, 8, 2, 3, 7, 12]
Hint: Array has an length of 6 sum of first half is 21 and sum of 2nd half in 22

```
public class SortArray {

    public static void main(String[] args) {

      Integer [] input = new Integer[] {2, 3, 4, 5};
       sortMyArray(input);
    }

    public  void  sortMyArray(Integer [] input) {
        //TODO Add your implementation here
        return;
    }
}
```

Problem 3

Correct the given equation by adding Parentheses in the right place

Example :

*Input : (8 + 2) * (2 -1 = 10*
*Output : (8 + 2) * (2 -1) = 10*

*Input : (3 * (3 + 1) - 3 * 2 + 1 = 19*
*Output : (3 * (3 + 1) - 3) * 2 + 1 = 19*

Hint: use wolframalpha in case of any doubts about the equation

```
public class BalanceTheBrackets {

    public static void main(String[] args) {

       String input = "(8 + 2) * (2 -1 = 10";
        balanceTheEq(input);
    }

    public  void  balanceTheEq(String input) {
        //TODO Add your implementation here
        return;
    }
}
```