Week -1 Programming Exercise: Counterfeit Coin Riddle

Assume you are given an array. Each element in the array corresponds to the weight of a coin in the treasure. All the coins have the same weight except one (the counterfeit coin). The counterfeit coin is lighter than the other coins. Your aim is to determine the position of the counterfeit coin in the array and how much lighter it is as compared to the other coins. For comparing weights, you can only use the given balance scale. You can put any number of coins on two sides of the scale. The balance scale can be implemented as a function which takes two arguments, 1) coins on the left side and 2) coins on the right side, and returns which side of the balance is lighter. If required, you can have more arguments in this function.

The command line arguments will be used to define the weights of the coins. The first argument is the number of coins in the treasure. Let the number of coins be k. The next k inputs correspond to the weight of the coins. Ensure proper sanity checks for correct input. Print "Wrong Input" for invalid input.

In this assignment, you will implement several algorithms to solve this riddle. At every step, your algorithm should output the weights placed on both the sides of the scale. The elements should be processed in the array order. The last line of the output should be the index of the counterfeit coin and the weight difference. Sample inputs and outputs have been shown for a few test cases.

Algorithm#1: Pairwise Comparison

Keep comparing the pair of elements one by one. As soon as you find an element with lesser weight, return its index and the weight difference.

Sample Input#1:
7 5 5 5 5 2 5 5
Output:
5 5
5 5
2 5
4 3

Sample Input#2:
5 5 5 5 5 2
Output:
5 5
5 5
4 3

Algorithm#2: Divide and Conquer Strategy (A)

At every step, divide the treasure in 2 halves. Keep the first half on the left side of the scale and second half on the right side. In case one coin is left out (when the number of coins is odd), keep it separately. If both the sides are equal, then the coin left outside is the counterfeit coin. Else collect the coins on the lighter side and repeat this whole process again.

Sample Input#1:
5 5 5 5 5 2
Output:
10 10
4 3

Sample Input#2:
7 5 5 5 5 2 5 5
Output:
15 12
5 2
4 3

Algorithm#3: Divide and Conquer Strategy (B)
At every step, divide the treasure in 3 parts. Assuming the number of coins to be k, the first and second part, each contains floor(k/3) coins. Remaining coins go to the third part. Keep the first part on the left side of the scale, second part on the right side and remaining coins outside. If the scale is balanced, follow the same process for the third part, else target the lighter side. Note: If k=2 at any step, simply compare.

Sample Input#1:
15 5 5 5 5 5 5 5 5 5 5 5 2 5 5
Output:
25 25
5 5
2 5
12 3

Sample Input#2:
10 5 5 5 5 5 5 5 5 5 2
Output:
15 15
5 5
5 2
9 3

The filename for the problems should be
[your name]_[your roll number]_pc.c
[your name]_[your roll number]_dca.c
[your name]_[your roll number]_dcb.c
respectively.

Takeaway (Note: These problems are just for thinking, not for evaluation)**:**

1. What if the counterfeit coin can be either lighter or heavier than the remaining coins.
2. What if you can place only a limited number of coins on the sides of the scale?