The cost of stock on each day is given in an array A[] of size N. Find all the days on which you buy and sell the stock so that in between those days your profit is maximum.

Example 1:

```
Input:
N = 7
A[] = {100,180,260,310,40,535,695}
Output:
(0 3) (4 6)
Explanation:
We can buy stock on day 0,
and sell it on 3rd day, which will
give us maximum profit. Now, we buy
stock on day 4 and sell it on day 6.
```

Example 2:

```
Input:
N = 5
A[] = {4,2,2,2,4}
Output:
(3 4)
Explanation:
We can buy stock on day 3,
and sell it on 4th day, which will
give us maximum profit.
```

Your Task:
The task is to complete the function **stockBuySell()** which takes an array A[] and N as input parameters and finds the days of buying and selling stock. The

function must return a 2D list of integers containing all the buy-sell pairs. If there is No Profit, return an empty list.

Expected Time Complexity : O(N)
Expected Auxiliary Space: O(N)

Constraints:
$2 <= N <= 10^3$
$0 <= Ai <= 10^4$