714 Best time to Buy and sell stock with transcation fee

Your are given an array of integers `prices`, for which the `i`-th element is the price of a given stock on day `i`; and a non-negative integer `fee` representing a transaction fee.

You may complete as many transactions as you like, but you need to pay the transaction fee for each transaction. You may not buy more than 1 share of a stock at a time (ie. you must sell the stock share before you buy again.)

Return the maximum profit you can make.

Example 1:

```
Input: prices = [1, 3, 2, 8, 4, 9], fee = 2
Output: 8
Explanation: The maximum profit can be achieved by:
```

- Buying at prices[0] = 1
- Selling at prices[3] = 8
- Buying at prices[4] = 4
- Selling at prices[5] = 9

The total profit is ((8 - 1) - 2) + ((9 - 4) - 2) = 8.

Intution :
we will maintain cash and hold
cash keeps information about total cash money we are having in hand(Required profit)
hold keeps information about total stock income we are having
now for each ith day
cash = max(cash,hold + prices[i] - 2); // Either i will sell or not
hold = max(hold,cash - prices[i]); // initially hold will be -prices[0], so you will never consider buying one before selling one
at the end **cash** will be your final profit

CODE:

```
int maxProfit(vector<int>& prices, int fee) {
    if(prices.size() <= 1)
        return 0;
    int n = prices.size();
    int cash = 0,hold = -prices[0];
    for(int i=0;i<n;i++){
        cash = max(cash, hold + prices[i] - fee);
        hold = max(hold,cash - prices[i]);
    }
        return cash;
    }
```