2.Medium\06.Stock_buy_sell.cpp

```
/*
QUESTION:-
You are given an array prices where prices[i] is the price of a given stock on the ith day.
You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock.
Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.

Example 1:

Input: prices = [7,1,5,3,6,4]
Output: 5
Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.
Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.

Example 2:

Input: prices = [7,6,4,3,1]
Output: 0
Explanation: In this case, no transactions are done and the max profit = 0.
*/

/*
APPROACH:-
Initialize two variables: min_price and max_profit.

-> min_price = minimum price in the array.
-> max_profit = 0.

Iterate through the array, and for each price:

-> Update min_price to the minimum price seen so far.
-> Update max_profit to the maximum profit seen so far, or the current price minus min_price, whichever is greater.

Return max_profit.
*/

// CODE:-
int maxProfit(vector<int> &prices)
{
    int minprice = prices[0];
    int ans = 0;
    for (int i = 1; i < prices.size(); i++)
    {
        ans = max(ans, prices[i] - minprice);
        minprice = min(minprice, prices[i]);
    }
    return ans;
}

```