**BEST TIME TO BUY AND SELL STOCK**

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.

**CODE**

class Solution {

public:

    int maxProfit(vector<int>& prices) {

        int min\_price = 10000;

        int max\_profit = 0;

        for (int i = 0; i < prices.size(); ++i)

        {

            int price = prices[i];

            min\_price = min(min\_price, price);

            int profit = price - min\_price;

            max\_profit = max(max\_profit, profit);

        }

        return max\_profit;

    }

};

**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.