

<https://course.acciojob.com/idle?question=df87917b-67af-48b0-a132-0832e02a38f8>

Buy and Sell Stock

You are given an array `prices` where `prices[i]` is the price of a given stock on the i th 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.

Print the maximum profit you can achieve from this transaction. If you cannot achieve any profit, print 0.

Input Format

The first line of the input contains the number n (length of `prices` array)

The next n integers denote the price of the stock on the i th day.

Output Format

Print the maximum profit you can achieve from this transaction. If you cannot achieve any profit, print 0.

Example 1

Input

```
6
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

```
5
7 6 4 3 1
```

Output

```
0
```

Explanation

In this case, no transactions are done and the max profit = 0.

Constraints

```
1 <= prices.length <= 10^5
```

```
0 <= prices[i] <= 10^4
```

Topic Tags

- **Math**

My code

```
// n java
import java.util.*;
import java.io.*;

public class Main {
```

```

public static void main(String args[]) {
    Scanner input = new Scanner(System.in);
    int n = input.nextInt();
    int a[] = new int[n];
    for(int i = 0; i < n; i++){
        a[i] = input.nextInt();
    }
    int ans = Solution.bestTimeToBuyAndSell(n,a);
    System.out.println(ans);
}
}

```

```

class Solution {
    static int bestTimeToBuyAndSell(int n, int[] arr) {
        int ans=0;
        int cp=arr[0];
        for(int i=0;i<arr.length;i++)
        {
            if(cp>arr[i])
                cp=arr[i];
            if((arr[i]-cp)>ans)
                ans=arr[i]-cp;
        }
        return ans;
    }
}

```