🎭

Solutions

DSA Lab Sheet 1

21 January, 2025

Problem 1

Problem Statement: Given a non-negative integer ***u*** as input, reverse the digits of ***u*** and print out the output. You may ignore leading zeros while reversing the digits.

Constraints: $0 \le u \le 10^9$

Input: The input is a single integer ***u*** - the number to be reversed.

Example 1	**Example 2**	**Example 3**	**Example 4**
Input: 9348 **Output:** 8439	**Input:** 12345678 **Output:** 87654321	**Input:** 100000 **Output:** 1	**Input:** 0 **Output:** 0

Solution to Problem 1

```
int reverse(int u) {
   int v = 0;  // Initialize the result variable

   while (u != 0) {
      v *= 10;       // Shift v to the left by 1 decimal place
      v += u % 10;   // Add the last digit of u to v
      u /= 10;       // Remove the last digit of u
   }

   return v;  // Output the reversed number
}
```

Problem 2

Problem Statement: 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.

Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return **0**.

Input: The first line contains and integer ***n*** integers ***prices[1], prices[2], …, prices[n]***.

Example 1	**Example 2**
Input: 7 1 5 3 6 4 **Output:** 5	**Input:** 7 6 4 3 1 **Output:** 0

Solution to Problem 2

```
int stock_price(int[] prices) {
    int n = prices.length;

    // Initialize the max_price array
    int[] max_price = new int[n + 1];
    max_price[n] = 0;

    // Fill the max_price array with the maximum prices from the end to the start
    for (int i = n - 1; i >= 0; i--) {
        max_price[i] = Math.max(prices[i], max_price[i + 1]);
    }

    // Calculate the maximum profit
    int profit = 0;
    for (int i = 0; i < n; i++) {
        profit = Math.max(profit, max_price[i] - prices[i]);
    }

    return profit;
}
```