**Problem: Next Greater Element**

*Note: This problem is an adaptation of one from Geeks for Geeks (https://www.geeksforgeeks.org/next-greater-element/). I encourage you to view their solution and identify how it differs from ours in the lectures.*

Write a program to find the **next greater element** for each array element. The next greater element for an element x is the first greater element to the right of x in the array. If there is no greater element, use -1 for that position.

**Input:**

An array of integers. For example: [4, 5, 2, 25].

**Output:**

An array where each element is replaced with the next greater element, or -1, if no such element exists.

**Example:**

Input: [4, 5, 2, 25]

Output: [5, 25, 25, -1]

Input: [13, 7, 6, 12]

Output: [-1, 12, 12, -1]

Using a simple linear approach from left to right and nested loops is not very difficult.

* Take 4, look right until you find the next value larger than it (5)
* Take 5, look right until you find the next value larger than it (25)
* Take 2, look right until you find the next value larger than it (25)
* Take 25, look right until you find the next value larger than it (-1)

**Challenge:** Use a stack and work backward through the array.

**Hints:**

* The stack helps keep track of potential next greater elements as you traverse the array.
* Think about it backward. By traversing from right to left, you ensure you always have access to the next elements as they appear on the right.
* It still needs some looping.

Input: [4, 5, 2, 25]

Look at the 25; the stack is empty, and there is no greater element, so -1, but 25 could be the next greater element, so push it to the stack.

Result: [0, 0, 0, -1]

Stack: [25]

Look at the 2; the stack is not empty, so you will pop off until you find a value larger than 2. In this case, the 25 is greater, so 25. It is possible for 2 to be the next greater element, so push it.

Result: [0, 0, 25, -1]

Stack: [25, 2]

Look at the 5; the stack is not empty, so you will pop off until you find a larger value than 5. In this case, we pop the 2 and see the 25. It is possible for 5 to be the next greater element, so push it.

Result: [0, 25, 25, -1]

Stack: [25, 5]

Look at the 4; the stack is not empty, so you will pop off until you find a larger value than 4. In this case, we see 5, which is larger. It is possible for 4 to be the next greater element, so push it.

Result: [5, 25, 25, -1]

Stack: [25, 5, 4]

Now, we are at the end of the array.

**Algorithm:**

1. Initialize an empty stack.
2. Traverse the array from **right to left**.
   * For each element, pop elements from the stack until you find a greater element or the stack becomes empty.
   * If the stack is not empty, the top of the stack is the next greater element.
   * If the stack is empty, there is no greater element, so the result for that position is -1.
   * Push the current element onto the stack before moving to the next element.
3. Store the results in an output array.