Replace Elements with Greatest Element on Right Side

# Question

Given an array arr, replace every element in that array with the greatest element among the elements to its right, and replace the last element with -1.

After doing so, return the array.

**Example 1:**

Input: arr = [17,18,5,4,6,1]

Output: [18,6,6,6,1,-1]

Explanation:

- index 0 --> the greatest element to the right of index 0 is index 1 (18).

- index 1 --> the greatest element to the right of index 1 is index 4 (6).

- index 2 --> the greatest element to the right of index 2 is index 4 (6).

- index 3 --> the greatest element to the right of index 3 is index 4 (6).

- index 4 --> the greatest element to the right of index 4 is index 5 (1).

- index 5 --> there are no elements to the right of index 5, so we put -1.

**Example 2:**

Input: arr = [400]

Output: [-1]

Explanation: There are no elements to the right of index 0.

# Pseudo Code

If Size of the Array is 0

Return

Allocate the Memory in the Heap for the array to be returned.

Declare and Initialize relativeMaxValue to 0

Run the Outer for loop (i = 0 to arrSize)

Run the Inner for loop (j = i + 1 to arrSize)

If relativeMaxValue < arr[j]

relativeMaxValue = arr[j]

arr[i] = relativeMaxValue

relativeMaxValue = 0

arr[arrSize - 1] = -1

# Source Code

## v 1.0 (Language C)

1. /\*\*
2. \* Note: The returned array must be malloced, assume caller calls free().
3. \*/
4. int\* replaceElements(int\* arr, int arrSize, int\* returnSize){
6. \*returnSize = arrSize;
7. int\* ptr = (int\*)calloc(arrSize, sizeof(int));
9. int relativeMaxValue = 0;
10. for(int i=0 ; i<arrSize ; i++) {
12. for(int j = (i+1) ; j<arrSize ; j++) {
14. if(relativeMaxValue < arr[j])
15. relativeMaxValue = arr[j];
16. }
17. ptr[i] = relativeMaxValue;
18. relativeMaxValue = 0;
19. }
20. ptr[arrSize - 1] = -1;
22. return ptr;
23. }