* Problem statement :-

→ Given are the two arrays arr1 & arr2 where each element is an integer.

→ Write a function that adds each digit of the array starting from its last position & returns the result array.

* Examples :-

arr1 = [9, 9, 9] arr2 = [9, 9, 9, 9]

result = [1, 0, 9, 9, 8]

```
    9 9 9
+ 1
  9 9 9 9
-------
1 0 9 9 8
```

arr1 = [1] arr2 = [9]

result = [1, 0]

* Solution :-

→ Create a list to store the result.

→ Set two pointers : One for the last position of arr1 & one for arr2.

→ Initialize carry = 0.

→ While at least one of the arrays still has digits left, or the carry is non-zero :

i) Initialize sum = 0 for each iteration.

ii) If pointer of arr1 has any element, then add it to the sum & decrement the pointer.

iii) If pointer of arr2 has any element, then add it to the sum & decrement the pointer.

iv) Add carry to sum.

v) Compute the last digit sum % 10

vi) Compute the carry sum / 10

v) Add the last digit to the 0^{th} index of result.

→ After the loop, if carry != 0, add carry at the 0^{th} index of result.

→ Convert result to int[] and return it.

Time complexity :- $O(n)$ where $n = \max(arr1.length, arr2.length)$

Space complexity :- $O(n)$ " " "