Exercise 1: Radix Sort Example

Sorting the following arrays using the Radix Sort algorithm:

Array 1: 34, 9134, 20134, 29134, 4, 134

Initial	Ones	Tens	Hundreds	Thousands	Ten Thousands
34	4	4	4	4	4
9134	34	34	34	34	34
20134	134	134	134	134	134
29134	9134	9134	9134	9134	9134
4	20134	20134	20134	20134	20134
134	29134	29134	29134	29134	29134

Final sorted array: 4, 34, 134, 9134, 20134, 29134

Array 2: 4, 34, 134, 9134, 20134, 29134

This array is already sorted, so each pass will maintain the same order.

Array 3: 29134, 20134, 9134, 134, 34, 4

Initial	Ones	Tens	Hundreds	Thousands	Ten Thousands
29134	29134	29134	4	4	4
20134	20134	34	34	34	34
9134	9134	9134	134	134	134
134	134	134	9134	9134	9134
34	34	20134	20134	20134	20134
4	4	29134	29134	29134	29134

Final sorted array: 4, 34, 134, 9134, 20134, 29134

Exercise 2: O(n) Sorting Algorithm for Positive Integers

Algorithm:

1. Choose base $k = n$

2. Convert all numbers from base 10 to base k

3. Apply Radix Sort to sort the numbers

Example (a):

Input: 45, 98, 3, 82, 132, 71, 72, 143, 91, 28, 7, 45
$n = 12$ (since $12^2 - 1 = 143$)

Base 10	Base 12	Least Significant	Most Significant	Final (Base 10)
45	39	60	03	3
98	82	70	07	7
3	03	B0	24	28
82	6A	03	39	45
132	B0	24	39	45
71	5B	39	5B	71
72	60	39	60	72
143	BB	79	6A	82
91	79	82	79	91
28	24	5B	82	98
7	07	6A	B0	132
45	39	BB	BB	143

Example (b):

Input: 45, 98, 3, 82, 132, 71, 72, 143, 91, 28, 7, 45, 151, 175, 145, 399, 21, 267, 346, 292

$n = 20$ (since $20^2 - 1 = 399$)

Base 10	Base 20	Least Significant	Most Significant	Final (Base 10)
45	25	11	03	3
98	4I	21	07	7
3	03	25	11	21
82	42	25	1I	28
132	6C	03	25	45
71	3B	73	25	45
72	3C	03	3B	71
143	73	42	3C	72
91	4B	82	42	82
28	1I	92	4B	91
7	07	4B	4I	98
45	25	6C	6C	132
151	7B	3B	73	143
175	8F	3C	75	145
145	75	75	7B	151
399	JJ	7B	8F	175
21	11	8F	D7	267
267	D7	D7	EC	292
346	H6	EC	H6	346
292	EC	H6	JJ	399

This algorithm runs in O(n) time because it performs a constant number of passes (2 in this case) over the n elements, regardless of the size of the numbers.