Exercise 1 — Insertion Sort

You are given an array of integers that must be sorted in ascending order using the **insertion sort** technique. The exercise is divided into **two parts**.

Part 1 – Insert the last element

You are given an array that is **already sorted**, except for its **last element**, which may be smaller than some of the previous ones.

Write pseudocode for an algorithm that **inserts the last element into its correct position** in the sorted part of the array. At each step:

1. Shift one element of the sorted part to the right if it is greater than the value to insert.
2. **Print the array** after each shift.
3. Once the correct position is found, insert the value and **print the array** again.

Example

Input:

```
2 4 6 8 3
```

Expected output:

```
2 4 6 3 8
2 4 3 6 8
2 3 4 6 8
2 3 4 6 8
```

Part 2 – Full insertion sort

Extend your previous algorithm so that it **sorts the entire array**.
In this version, you must repeatedly apply the same insertion idea to each element of the array (starting from the second element).
After each insertion step, **print the whole array** to show its intermediate state.

Example

Input:

```
1 4 3 5 6 2
```

Expected output:

```
1 4 3 5 6 2
1 3 4 5 6 2
1 3 4 5 6 2
1 3 4 5 6 2
1 2 3 4 5 6
```

What to deliver

- Pseudocode for both parts (the algorithm of the first part can be used as a subroutine invoked in the second part).
- A brief explanation of how the algorithm works (also with comment in the pseudocode)
- The time complexity for each part.

Exercise 2 — Recursive Palindrome

A **palindrome** is a string that reads the same forward and backward (for example, `"radar"`, `"level"`, `"abba"`).

In this exercise, you are asked to write pseudocode for an algorithm that **checks whether a given string is a palindrome**, using a **recursive approach**.

Definition

A string is a palindrome if:

- It contains **fewer than two characters** (base case), or
- Its **first and last characters are the same**, and the **substring obtained by removing them** is also a palindrome (recursive case).

Task

Write pseudocode for a recursive algorithm `isPalindrome(s)` that returns `True` if the string `s` is a palindrome, and `False` otherwise.

You may assume that the string:

- Contains only lowercase letters, and
- Has no spaces or punctuation.

Example

Input:

```
radar
```

Output:

```
True
```

Input:

```
hello
```

Output:

```
False
```

Requirements

- Clearly define the **base case** and **recursive case** in your pseudocode.
- Explain in a few lines **why** your recursion works (i.e., why it eventually terminates and correctly identifies palindromes).
- Indicate the **time complexity** of your algorithm as a function of the string length.

Exercise 3 — Priyanka and Toys

Priyanka works for an international toy company that ships its products in **containers**. Each container can only hold items whose **weights are within 4 units** of the **lightest item** placed inside it.

Given a list of item weights, Priyanka wants to determine the **minimum number of containers** needed to ship all items.

Task

Write pseudocode for an algorithm that, given a list of integers `w` representing item weights, returns the **minimum number of containers** required so that every item can be shipped under the rule:

> For each container, all items inside must have weight ≤ (minimum weight in that container + 4).

Example

Input:

```
1 2 3 21 7 12 14 21
```

Output:

```
4
```

Explanation

- The first container holds items with weights `1, 2, 3` (range 1–5).
- The second container holds `7` (range 7–12).
- The third container holds `12, 14` (range 13–18).
- The fourth container holds `21, 21` (range 13–18 and 21–25; note that 21 requires a new container).
  → **4 containers** are required.

Requirements

- Clearly describe the algorithm in pseudocode.
- Indicate and justify the **complexity** of your approach.