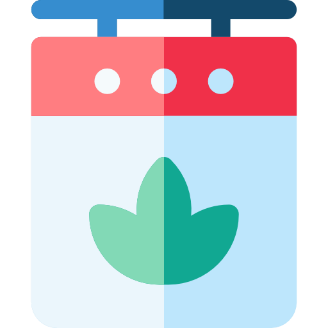
**Advent Calendar**

**

*While Santa is delivering all the presents to the kids, the parents are buing and hanging some advent calenedars for the kids to enjoy. While buying the advent calendars, most of the families saw that the numbers in the calendars are kind of messed up, so it is your job to fix them.*

Create a function called **fix\_calendar**. The function should receive some **shuffeled numbers** (the days on the calender) and it should **return** them correctrly **ordered (ascening)**. The numbers passed to your function will always be **positive**. You are **not allowed** to:

* Use any of the **built-in functions** to sort the numbers.
* Create **new lists** to help you
* **Delete items** from the given list

***Note: Submit only your function in the judge system.***

### Input

* There will be **no input**
* You can **test your code** using your own **examples** or those given below

### Output

* Your function should return a **list of the sorted** numbers in **ascending order**

### Examples

|  |  |  |
| --- | --- | --- |
| **Test Code** | **Output** | **Comments** |
| numbers = [3, 2, 1]  fixed = fix\_calendar(numbers)  print(fixed) | [1, 2, 3] | We start by comparing **3** with **2** -> **3 > 2**, so we **swap** them: **[2, 3, 1]**  We compare **3** with **1** -> **3 > 1**, so we **swap** them: **[2, 1, 3]**  We checked all the numbers, but there were swaps, so we **loop again**  We copare **2** with **1** -> **2 > 1**, so we **swap** them: **[1, 2, 3]**  We compare **2** with **3** -> **2 < 3**, **no swap**  We checked all again, there was a swap, so we **loop again**  This time we loop throug all of them with **no swaps**, so we have **sorted** the numbers |