

# 1. 字串重新排列

---

## 說明

給你一個字符串 `s` 和一個 **長度相同** 的整數數組 `indices`。

請你重新排列字符串 `s`，其中第 `i` 個字符需要移動到 `indices[i]` 指示的位置。

返回重新排列後的字符串。

### 示例 1:

- 輸入: `s = "techcheer"`, `indices = [5, 3, 0, 1, 7, 8, 6, 2, 4]`
- 輸出: "cheertech"

### 示例 2:

- 輸入: `s = "abc"`, `indices = [0, 1, 2]`
- 輸出: "abc"

### 約束條件:

- `s.length == indices.length == n`
- `1 <= n <= 100`
- `s` 由小寫英文字母組成。
- `0 <= indices[i] < n`
- `indices` 的所有值都是**唯一的**。

## 2. 重新排列數組

---

### 說明

給你一個數組 `nums`，數組中有 `2n` 個元素，按 `[x1,x2,...,xn,y1,y2,...,yn]` 的格式排列。

請你將數組按 `[x1,y1,x2,y2,...,xn,yn]` 格式重新排列，返回重排後的數組。

示例 1:

- 輸入: `nums = [2,5,1,3,4,7]`, `n = 3`
- 輸出: `[2,3,5,4,1,7]`
- 解釋: 由於 `x1=2`, `x2=5`, `x3=1`, `y1=3`, `y2=4`, `y3=7`，所以答案為 `[2,3,5,4,1,7]`

示例 2:

- 輸入: `nums = [1,2,3,4,4,3,2,1]`, `n = 4`
- 輸出: `[1,4,2,3,3,2,4,1]`

示例 3:

- 輸入: `nums = [1,1,2,2]`, `n = 2`
- 輸出: `[1,2,1,2]`

約束條件:

- `1 <= n <= 500`
- `nums.length == 2n`
- `1 <= nums[i] <= 10^3`

### 3. 刪除子串後的字符串最小長度

#### 說明

給你一個僅由 大寫 英文字符組成的字符串 `s` 。

你可以對此字符串執行一些操作，在每一步操作中，你可以從 `s` 中刪除 任一個 `"AB"` 或 `"CD"` 子字符串。

通過執行操作，刪除所有 `"AB"` 和 `"CD"` 子串，返回可獲得的最終字符串的 最小 可能長度。

**注意**，刪除子串後，重新連接出的字符串可能會產生新的 `"AB"` 或 `"CD"` 子串。

**示例 1:**

- **輸入:** `s = "ABFCACDB"`
- **輸出:** `2`
- **解釋:**

你可以執行下述操作：

- 從 `"ABFCACDB"` 中刪除子串 `"AB"`，得到 `s = "FCACDB"` 。
- 從 `"FCACDB"` 中刪除子串 `"CD"`，得到 `s = "FCAB"` 。
- 從 `"FCAB"` 中刪除子串 `"AB"`，得到 `s = "FC"` 。

最終字符串的長度為 `2` 。

可以證明 `2` 是可獲得的最小長度。

**示例 2:**

- **輸入:** `s = "ACBBD"`
- **輸出:** `5`
- **解釋:** 無法執行操作，字符串長度不變。

**約束條件:**

- `1 <= s.length <= 100`
- `s` 僅由大寫英文字母組成

## 4. 查找共用字符

---

### 說明

給你一個字符串數組 `words`，請你找出所有在 `words` 的每個字符串中都出現的共用字符（**包括重複字符**），並以數組形式返回。你可以按 **任意順序** 返回答案。

示例 1:

- 輸入: `words = ["bella","label","roller"]`
- 輸出: `["e","l","l"]`

示例 2:

- 輸入: `words = ["cool","lock","cook"]`
- 輸出: `["c","o"]`

約束條件:

- `1 <= words.length <= 100`
- `1 <= words[i].length <= 100`
- `words[i]` 由小寫英文字母組成