

陣列操作挑戰

問題描述：

本題會提供一個二維 array (mode 1) 或一個二維 vector (mode 2)，並根據不同的操作指令進行陣列操作。

輸入說明：

首先輸入一個整數 mode，表示操作模式：

- mode == 1：對一個 3x3 的二維陣列進行操作，此陣列初始化為：

```
int array2d[3][3] = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8}};
```

- mode == 2：對一個 len*len 的二維向量進行操作。

mode 1:

- 輸入 3 個整數 op，每個 op 代表一個操作指令：
 - op == 1：將陣列中所有元素加 1。
 - op == 2：將陣列中間那列 (row index 1) 的所有元素設為 3。
 - op == 3：將陣列中所有元素對 3 取餘數。

mode 2:

- 輸入一個整數 len，表示二維向量的尺寸為 len*len。
- 輸入 len*len 個整數，依序填入二維向量。
- 初始化 `vector<vector<int>> v(len, vector<int>(len, 0));`
- 輸入 len 個整數 op，每個 op 代表一個操作指令：
 - op == 1：將整個二維向量向上推一格 (push_up)。
 - op == 2：將整個二維向量向下推一格 (push_down)。
 - op == 3：將整個二維向量向左推一格 (push_left)。
 - op == 4：將整個二維向量向右推一格 (push_right)。

注意事項：

- push_up 操作：將第一列移到最後一列，其餘列向上移動。
- push_down 操作：將最後一列移到第一列，其餘列向下移動。
- push_left 操作：將每一列的第一個元素移到最後，其餘元素向左移動。
- push_right 操作：將每一列的最後一個元素移到最前面，其餘元素向右移動。

輸出說明：

- mode 1: 輸出操作後的 3x3 陣列，不用空格
- mode 2: 輸出操作後的二維向量，不用空格
- 測資中元素操作結果範圍：整數 0~9，len 範圍：整數 3~10，mode 輸入只會有 1、2

範例測資：

Sample1 Input	Sample1 Output
1 3 2 1	123 444 123
Sample2 Input	Sample2 Output
2 3 1 2 3 4 5 6 7 8 9 2 2 4	645 978 312

繳交格式：

1. 上傳內容須為 zip 檔
 - 程式碼(請繳交.cpp 檔)
 - 檔名為 AXX-學號 (Ex：A06-113502000) (XX：第幾份作業)
 - 直接對.cpp 檔壓縮即可，不需要在外面再加一層資料夾
2. 請針對程式內的重要功能進行「適當註解」，簡單說明該段程式碼的作用，註解的部分會納入評分考量，請勿隨意註解。
3. 記得在 Online Judge 系統上測試，若在正常繳交期限前未有測試紀錄，將扣該次作業 10 分。
4. 若檔名錯誤，扣該次作業 10 分；若沒壓縮，扣該次作業 10 分。

繳交期限：

- 2024/10/24 23:59 前將檔案上傳到 ee-class 上

遲交期限：

- 2024/10/27 23:59 前將檔案上傳到 ee-class 上
- 該次作業分數打八折
- 遲交期限過後不再開放繳交作業，該次以 0 分計算