

Homework 5

貨船調度



圖為歐洲最大的港口，位於荷蘭的鹿特丹港

題目:

假設港口是一個堆疊，所有貨船(在程式中以數字表示)都要入港一次，先進來的貨船要等待後面進來的貨船出港後才能離開。這時我們有一個函式 `dispatch`，它有兩個參數，一個是 `ship_in`，代表貨船入港順序，另一個是 `ship_out`，代表貨船出港順序。這個函式的功能是要根據入港順序與出港順序，列出所有貨船出入港的程序(在函式中為一個名為 `step` 的串列)，出港和入港分別用 `pop` 和 `push` 表示。

例如，假設今天貨船的入港順序是 `[1, 2, 3]`，出港順序是 `[2, 1, 3]`，整個貨船出入港的程序為：

1號船入港 (`push`) → 2號船入港 (`push`) → 2號船出港 (`pop`，2號船目前為最後進來的船，也是最先出港的船) → 1號船出港 (`pop`) → 3號船入港 (`push`) → 3號船出港 (`pop`)

所以最後 `step` 串列的結果為 `['push', 'push', 'pop', 'pop', 'push', 'pop']`。

另一個例子: 假設今天貨船的入港順序是[1, 2, 3], 出港順序是[3, 1, 2], 整個貨船出入港的程序為:

1號船入港 (push) → 2號船入港 (push) → 3號船入港 (push) → 3號船出港 (pop)

此時下一個要出港的是1號船, 但是1號船前面還有2號船, 使得1號船無法出港, 所以這個情況我們只回傳空的step串列。

請根據以上解釋, 完成dispatch函式, 這個函式要回傳step串列並在主程式列出結果。

提示:

當即將出港的船(可以想成堆疊的頂端)和出港順序串列的第一項是一樣的, 則會把pop放進step裡面。

當遇到出入港是同一艘船的情況, 則會把push與pop一起放進step裡面。

範例1

輸入:

```
ship_in = [1,2,3]
ship_out = [3,1,2]
```

輸出:

```
[]
```

範例2

輸入:

```
ship_in = [1, 5, 8, 6, 9]
ship_out = [8, 5, 6, 1, 9]
```

輸出:

```
['push', 'push', 'push', 'pop', 'pop', 'push', 'pop', 'pop', 'push', 'pop']
```

完成程式後請將程式命名為hw5_ship_dispatch_學號.py