

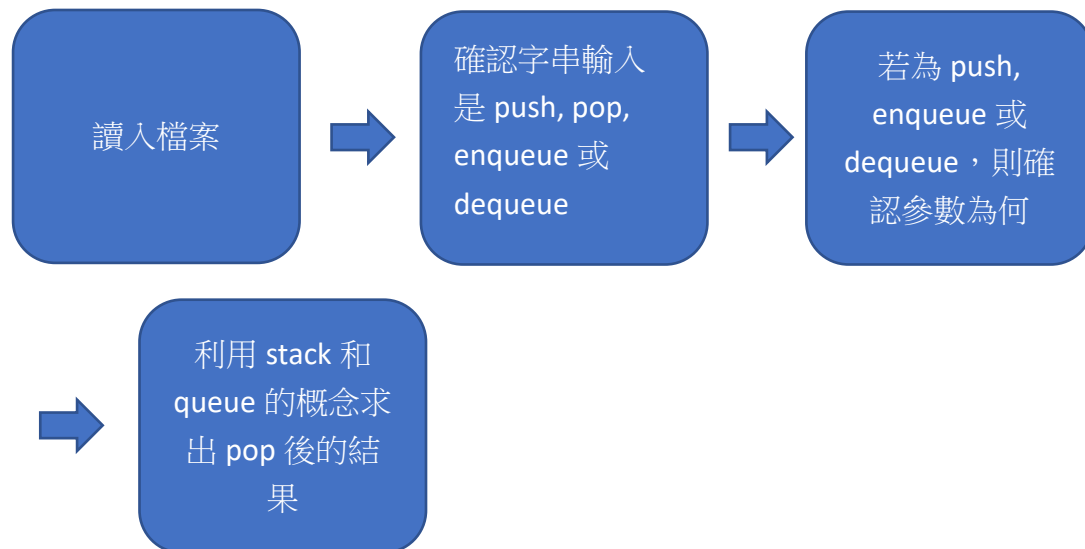
## 一、執行結果

```
PS D:\Users\User\Desktop\DS_HW2> gcc -o hw2 hw2.c
PS D:\Users\User\Desktop\DS_HW2> ./hw2
```



54  
98  
3  
1  
30

## 二、流程圖



## 三、函式說明

**void push(int data)**

- **top** 原本在-1，代表空 **stack**。Push 一個資料進來後，**top** 就往上疊，跑到 0，同時在那個位置儲存資料。當 **top** 到達最大容量-1 時(因為以 0 作為起點，故最後一個位置是最大容量-1)，就印出 **stack** 已滿的訊息。

**int pop()**

- 先用一個變數儲存目前位於 **top** 的資料(也就是待會要被 **pop** 的資料)，接著讓 **top** 減 1，無視掉原 **top** 的資料，視同將它 **pop** 出去。最後將儲存原 **top** 資料的變數回傳，即得想要結果。

**void enqueue(char data)**

- 一開始 **front** 和 **rear** 都是 -1，當 **enqueue** 一個資料進來，**rear** 就加 1 跑到 0，同時在該位置儲存資料。當 **rear** 達到最大容量-1 時，印出 **queue** 已滿。

**char dequeue()**

- 先將 **front** 加 1，用一個變數儲存新 **front** 位置的資料(即為等下要被 **dequeue** 的資料)，再將變數回傳。

#### 四、設計討論

熟悉 **stack** 和 **queue** 的概念後其實就可以了。但我的程式只允許 **stack** 和 **queue** 固定大小，若能用指標和 **malloc** 的方式，就能改成動態分配空間了。同時我需要完善字串比對的方式，應該有除了一個個字元比更好的方法。再來就是傳的參數目前只能小於 **100**，如果能更熟悉如何將字串轉整數，就能增加它的包容性了。