1. **利用陣列實作大小為4的STACK (FILO 先進後出)，**

**使其有PUSH、POP功能，**

**PUSH，pop以字串輸入，並以string.h 中的strcmp()比對**

**PUSH : 輸入push 字串後面接要PUSH的值，放入STACK**

**POP : 將STACK最上面的值刪掉，TOP-1**

**當堆疊資料為空，則印出Stack Empty!**

**當堆疊資料滿了，則印出Stack Full**

**在每一次PUSH，POP的過程中，將堆疊裡面資料印出來**

**(堆疊大小麻煩設4，方便助教檢查)**

1. **以我們第一題做的堆疊ADT運用在第二題**

**讀取後序(Postfix Expression)加減乘除的式子**

**運算子(Operator): + - \* /**

**運算元(Operand): 0-9**

**以老師上課所教的堆疊演算法印出運算結果**

**若讀取時發生錯誤**

**(如非後序、運算元匹配數量不相符的運算子、出現非運算子運算元的東西…)**

**則印出Input Error !!**

**(堆疊大小請設15)**

**#ifndef** STACKLIB\_H\_

**#define** STACKLIB\_H\_

**#include** <stdio.h>

**#include** <stdlib.h>

**#define** MAX 4

**typedef** **struct** {

**int** value;

} element;

**void** **createStack**(element stack[]);

//將Stack初始化並Create

**int** **IsFull**(**int** top);

//判斷堆疊是否滿了

**int** **IsEmpty**(**int** top);

//判斷堆疊是否空了

**int** **push**(element stack[], **int**\* top, **int** value) ;

//堆疊放入Stack，top往上移

**int** **pop**(element stack[], **int**\* top) ;

//堆疊丟出Stack，top往下移 回傳值int為第二題，要return數值

**void** **listAll**(element stack[], **int**\* top);

//堆疊全部印出來

**#endif** /\* STACKLIB\_H\_ \*/

1. **Input**

push 9

push 5

push 2

push 7

push 3

pop

pop

pop

push 6

pop

pop

pop

**Output**

9

9 5

9 5 2

9 5 2 7

Stack Full

9 5 2

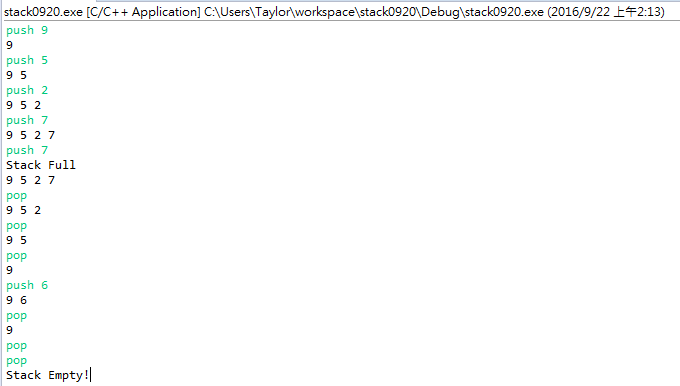
9 5

9

9 6

9

Stack Empty!

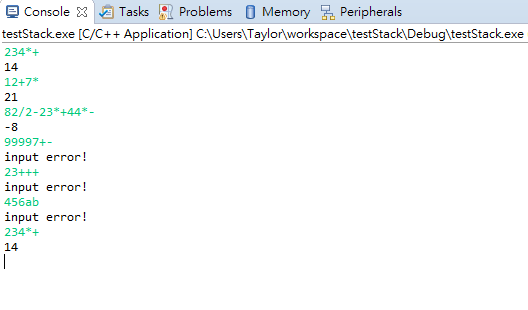


**Input**

1. 234\*+
2. 12+7\*
3. 82/2-23\*+44\*-
4. 9997+-
5. 23+++
6. 456ab

**Output**

1. 14
2. 21
3. -8
4. Input Error
5. Input Error
6. Input Error



**助教改作業時注意一下，出現input error後，前面測資可以在檢查看看，**

**有些人error可能會沒歸0**