

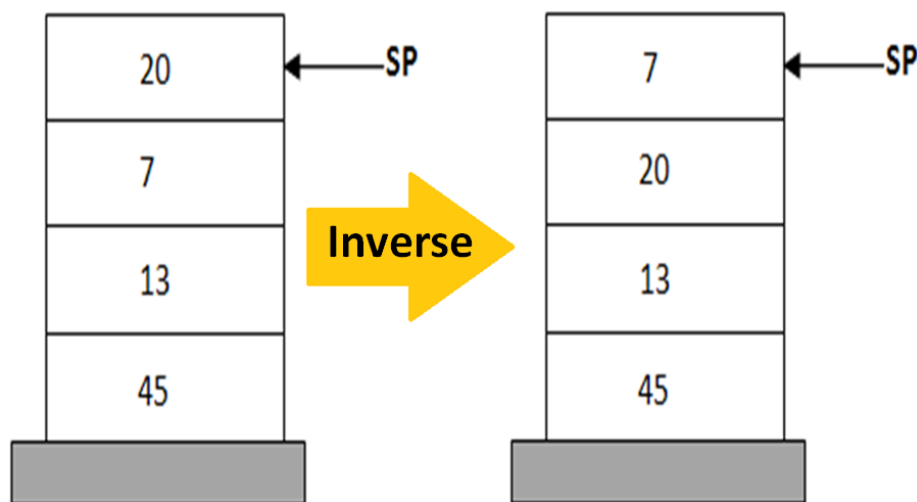
A. Stack-Inverse (50pt)

Description

注意：此題不限 C++, Yacc, Lex

一般的 Stack 有 push、pop、top、size 指令，此題目需要實作三種新的 Stack 指令，以及與一般的 Stack 有相同功能的 push 指令，這三種新指令分別是 **inverse**、**dec**、**inc**，你需要判斷這四種指令並實現相對應的功能，且保留 Stack 先進後出的特性，push 指令會放新的值到頂端，inverse 指令需要將最上方的兩個值做對調，dec 會取出最頂端的值並進行減 1 後放回去，inc 會取出最頂端的值並進行加 1 後放回去。

完成 4 個 push 指令後執行 inverse 指令的示意圖與測資輸入會呈現如下。



```
push 45
push 13
push 7
push 20
inverse
$
```

Input Format

每筆輸入為一連串指令，請依照指令進行相對應的動作。

1. 每筆輸入由 **\$** 做結尾。
2. 每個指令的文法語意都是正確的，不須考慮 syntax error 和 semantics error。
3. “inverse” 指令只會在 Stack 至少儲存了兩個值之後出現。

指令	描述
push	會空一格後面再接數字，並將後面的數字 push 到 Stack 上面
inverse	取出 Stack 最上層的兩個值，並互換位置 push 回去
inc	取出 Stack 最上層的 1 個值，並加 1 後放回去
dec	取出 Stack 最上層的 1 個值，並減 1 後放回去

Output Format

執行一連串的指令後，輸出 Stack 最頂端的值。

<p>Sample Input</p> <pre>push 1 push 5 dec inverse push 70 inverse inc push 20 inverse \$</pre> <p>Sample Output</p> <pre>2</pre>	<p>Sample Input</p> <pre>push 78 push 35 push 900 dec inverse inc push 20 inc push 3 inverse inverse \$</pre> <p>Sample Output</p> <pre>3</pre>
<p>Sample Input</p> <pre>push 2 push 18 inverse push 66 push 59 dec push 3 inverse push 78 inverse \$</pre> <p>Sample Output</p> <pre>58</pre>	<p>Sample Input</p> <pre>push 25 push 95 inc inverse push 152 inverse \$</pre> <p>Sample Output</p> <pre>25</pre>

提醒：如果用 C++ 做題目，有需要使用 C++ Stack 的人，可參考以下：

標頭檔

```
#include<stack> //標頭檔
```

宣告和使用函數

```
stack<int> sta;//宣告一個專門存放 int 的 stack  
int i=45;//宣告要存放的元素  
sta.empty();//回傳 stack 是否為空  
sta.size(); //回傳 stack 內堆疊了幾個元素  
sta.push(i);//將 45 放到 stack 最頂端  
i=sta.top(); //取得 stack 最頂端的元素 45，如果 stack 內沒有東西則會回傳 error  
sta.pop(); //移除 stack 最頂端的元素 45，如果 stack 內沒有東西則會回傳 error
```