

HW4 使用說明

總共分為三個頁面，為 infix ,prefix ,postfix

Infix :

輸入：infix (可以自行撰寫)

輸出：postfix

Prefix

預設部分：可自行選擇，複製貼上到輸入

The screenshot shows a Windows application window titled "Form1". At the top, there are three tabs: "infix", "prefix", and "postfix", with "infix" currently selected. Below the tabs, a text box contains the expression "A/B-(C+D)*E+A*C". To the right of this text box is a blue box labeled "輸入部分" (Input part) with an arrow pointing to the text box. Above the text box, a small note reads "Only binary operators in {+,-,*,/,^,and,|} are considered." Below the text box, there are two buttons: "infix => postfix" and "infix => prefix". To the right of these buttons is a red box labeled "輸出部分" (Output part) with an arrow pointing to the output area. Below the buttons, there is a "test cases:" label and a dropdown menu showing "Choose one". To the right of the dropdown is a yellow box labeled "預設部分" (Default part) with an arrow pointing to the dropdown. At the bottom of the window, there are three checkboxes: "print stack", "clear all", and "Delete all". Below these is a text area labeled "Memo1".

prefix :

輸入：prefix (可以自行撰寫)

輸出：postfix

infix

預設部分：可自行選擇，複製貼上到輸入

Form1

infix prefix postfix

Only binary operators in {+,-,*,/,^,and,|} are considered.

+-/AB*+CDE*AC

prefix => postfix

prefix => infix

test cases: Choose one

☐ print stack ☐ clear all

Memo1

輸入部分

輸出部分

預設部分

postfix :

輸入：postfix (可以自行撰寫)

輸出：prefix

infix

預設部分：可自行選擇，複製貼上到輸入

Form1

infix prefix postfix

Only binary operators in {+,-,*,/,^,and,|} are considered.

AB/CD+E*-AC*+

postfix => prefix

postfix => infix

test cases: Choose one

☐ print stack ☐ clear all

Memo1

輸入部分

輸出部分

預設部分

中間 panel

1. 勾選欄：列出執行過程中的 Stack 內容物
2. 勾選欄：產生新的結果時，消除之前的結果
3. 按鈕：一次清除所有結果

Form1

infix prefix postfix

Only binary operators in {+,-,*,/,^,and,|} are considered.

AB/CD+E*-AC*+

postfix => prefix

postfix => infix

test cases: Choose one

☐ print stack ☐ clear all Delete all

Memo1

下方 memo1 :

1. 輸出上方執行過後的結果
2. 過程中列出 Stack1 和 Stack2 的內容物

Form1

infix prefix postfix

Only binary operators in {+,-,*,/,^,and,|} are considered.

A/B-(C+D)*E+A*C

infix => postfix

infix => prefix

test cases: Choose one

☐ print stack ☐ clear all Delete all

Memo1

infix : A/B-(C+D)*E+A*C
postfix : AB/CD+E*-AC*+
infix : A/B-(C+D)*E+A*C
prefix : +/AB*+CDE*AC

Form1

infix prefix postfix

Only binary operators in {+,-,*,/,^,and,|} are considered.

+ /AB*+CDE*AC

prefix => postfix

prefix => infix

test cases: Choose one

☐ print stack ☐ clear all Delete all

Memo1

prefix : + /AB*+CDE*AC
postfix : AB/CD+E*-AC*+
prefix : + /AB*+CDE*AC
infix : A/B-(C+D)*E+A*C

Form1

infix prefix postfix

Only binary operators in {+,-,*,/,^,and,|} are considered.

AB/CD+E*-AC*+

postfix => prefix

postfix => infix

test cases: Choose one

☐ print stack ☐ clear all Delete all

Memo1

postfix : AB/CD+E*-AC*+
prefix : + /AB*+CDE*AC
postfix : AB/CD+E*-AC*+
infix : A/B-(C+D)*E+A*C

Form1

infix prefix postfix

Only binary operators in {+,-,*,/,^,and,|} are considered.

A/B-(C+D)*E+A*C

infix => postfix

infix => prefix

test cases: Choose one

☒ print stack ☐ clear all Delete all

Memo1

Stack 2 : A - /AB*+CDE
Stack 1 : + #
Stack 2 : C A - /AB*+CDE
Stack 1 : + #
Stack 2 : *AC - /AB*+CDE
Stack 1 : #
Stack 2 : + /AB*+CDE*AC
Stack 1 :
Stack 2 : + /AB*+CDE*AC
infix : A/B-(C+D)*E+A*C
prefix : + /AB*+CDE*AC