

HW10 document

40947007S 資工113 張懷齡

執行

```
g++ hw10.cpp  
./a.out
```

輸出、輸入介面

輸入介面：

```
siang@zhangxianglingdeMacBook-Pro data structure % g++ hw10.cpp  
siang@zhangxianglingdeMacBook-Pro data structure % ./a.out
```

```
-----  
Data Structure Homework 10  
40947007S 資工113 張懷齡  
-----
```

```
Enter the infix expression : █
```

輸入infix expression

提供的輸入偵錯(額外功能)：

1. 左右括弧不對稱
2. 加減乘除不等於數字-1
3. 輸入長度不能大於20
4. 輸入不合法的字元
5. 輸入中間出現空白鍵

```

Enter the infix expression : (A/B)*((C+D)-(E+G))
The postfix expression: AB/CD+EG+-*
The prefix expression: */AB-+CD+EG
The level-order of expression tree: */-AB++CDEG
一個一個輸入對應之數值：
A: 2
B: 2
C: 2
D: 2
E: 2
G: 2
Result of expression evaluation : 0
輸入Esc鍵結束程式，還未要結束按任意鍵繼續.....
^[

```

接下來功能四是要給不同變數輸入數值，這裡會要一個一個輸入，輸入的順序是照postfix出現的變數(A,B,C....)順序。

如果還要繼續下一組輸入，就不要輸入esc鍵，輸入任意鍵（例如：數字1,2,3.....）

最後結束時要輸入`^[]` 也就是esc鍵，注意，按完esc鍵要按enter輸入。

輸出說明

輸入infix後，會依序輸出

postfix expression

prefix expression

the level-order of expression

接著一個一個輸入對應的值後，會輸出計算後的結果

如果還要繼續下一組輸入，就不要輸入esc鍵，輸入任意鍵（例如：數字1,2,3.....）

函式功用

```

void push(char oper)
-> 用於把infix轉為postfix的stack
char pop()
-> 用於把infix轉為postfix的stack
int priority(char alpha)
-> 優先權
string convert(string infix)
-> 把infix轉為postfix

```

```

struct Node* construct(char inorder[], int start, int end,
                      char postorder[], int *pIndex)

-> 建二元樹
struct Node* constructTree(char inorder[], char postorder[], int n)
-> 建二元樹
void preorderTraversal(struct Node* root)
-> preorder走訪
void level_order (struct Node* ptr)
-> level order走訪
void addq(int front, int *rear, struct Node* item)
-> 用於level order走訪的queue
struct Node* deleteq(int *front, int rear)
-> 用於level order走訪的queue
void push2(int item)
-> 用於計算功能四postfix結果的stack
int pop2()
-> 用於計算功能四postfix結果的stack
void EvalPostfix(int postfix[], int n)
-> 計算功能四的結果

```

時間複雜度

n：為樹的節點數

m：the number of tokens in the expression

功能1 - 輸入一串 infix 運算式，輸出其 postfix 運算式： $O(m)$

功能2 - 輸出對應 prefix 運算式： $O(n)$

功能3 - 輸出對應二元運算樹之 level order 走訪順序： $O(n)$

功能4 - 輸出運算式計算結果： $O(n)$