HW1

助教：

陳奕侑 B11030010@mail.ntust.edu.tw

唐于硯 B11130006@mail.ntust.edu.tw

徐欣儀 B11130008@mail.ntust.edu.tw

陳彥儒 B11115004@mail.ntust.edu.tw

作業規定

- 分數：7%, (HW1-1: 3%, HW1-2: 4%)
- 作業密碼 : 09301014
- OJ 網址: https://nlp.csie.ntust.edu.tw:2021/contest
- OJ 截止日期：2024/~~10/14~~ 10/21 10:00 (截止前無限制上傳次數)
- 請將上傳到OJ的程式碼壓縮成 zip 檔，並命名為「學號_姓名.zip」 (例如：b1234567890_王小明.zip)上傳至 Moodle，遲交 0 分。
- 程式語言開放使用 C (gcc 5.4)、C++ (g++ 5.4)，除了標準輸入輸出 (例如：stdio.h)和字串 (例如：string.h)相關的 library，請勿使用其他 library，請自行實作基礎資料結構。
- 請務必注意作業時限，超過時限或沒有在moodle和OJ都繳交作業者皆以0分計算！
- 請勿抄襲他人程式碼

HW1-1 Bitwise Calculator

Description

Given a bitwise operation in infix form, output its postfix form and the top of the stack after each character is scanned.

Note:

1. The operands are single-digit non-negative numbers and should be saved in a variable of type ***char***

2. The operator precedence is identical to that of C/C++ compilers (**~** > **&** > **^** > **|**).

Test case A:
6|9^4&5
= 6|(9^(4&5))

Output:
6945&^|
6 9 4 5 4 13 15

Test case B:
7&~(4^5)|~(3|~2)
= (7&(~(4^5)))|(~(3|(~2)))

Output:
745^~&32~|~|
7 4 5 1 -2 6 3 2 -3 -1 0 6

Bitwise operation

6|(9^(4&5))

110 or (1001 xor (100 and 101))
110 or (1001 xor 100)
110 or (1101)
1111

7&~(4^5)|~(3|~2)

111 and not (100 xor 101) or not (011 or not 010)
111 and not (001) or not (111)
111 and 110 or 000
110 or 000
110

Algorithm to Convert Infix to Postfix.

```
Step 1: Add ")" to the end of the infix expression
Step 2: Push "(" on to the stack
Step 3: Repeat until each character in the infix notation is scanned
       IF a "(" is encountered, push it on the stack
       IF an operand (whether a digit or a character) is encountered, add it to the
       postfix expression.
       IF a ")" is encountered, then
         a. Repeatedly pop from stack and add it to the postfix expression until a
            "(" is encountered.
         b. Discard the "(". That is, remove the "(" from stack and do not
            add it to the postfix expression
       IF an operator O is encountered, then
         a. Repeatedly pop from stack and add each operator (popped from the stack) to the
            postfix expression which has the same precedence or a higher precedence than O
         b. Push the operator O to the stack
       [END OF IF]
Step 4: Repeatedly pop from the stack and add it to the postfix expression until the stack is empty
Step 5: EXIT
```

Precedence	Operator
1	~
2	&
3	^
4	|

https://en.cppreference.com/w/c/language/operator_precedence

Infix	Stack	Postfix
7&~(4^5)|~(3|~2))	(	7
7&~(4^5)|~(3|~2))	(&	7
7&~(4^5)|~(3|~2))	(&~	7
7&~(4^5)|~(3|~2))	(&~(	7
7&~(4^5)|~(3|~2))	(&~(	74
7&~(4^5)|~(3|~2))	(&~(^	74
7&~(4^5)|~(3|~2))	(&~(^	745
7&~(4^5)|~(3|~2))	(&~	745^
7&~(4^5)|~(3|~2))	(|	745^~&
7&~(4^5)|~(3|~2))	(|~	745^~&
7&~(4^5)|~(3|~2))	(|~(	745^~&
7&~(4^5)|~(3|~2))	(|~(	745^~&3
7&~(4^5)|~(3|~2))	(|~(|	745^~&3
7&~(4^5)|~(3|~2))	(|~(|~	745^~&3
7&~(4^5)|~(3|~2))	(|~(|~	745^~&32
7&~(4^5)|~(3|~2))	(|~	745^~&32~|
7&~(4^5)|~(3|~2))		745^~&32~|~|

Evaluation of Postfix Expression

```
Step 1: Add a ")" at the end of the
        postfix expression
Step 2: Scan every character of the
        postfix expression and repeat
        Steps 3 until ")"is encountered
Step 3: IF an operand is encountered,
        push it on the stack
        IF an operator O is encountered, then
        a. Pop the top two elements from the
           stack as A and B
        b. Evaluate B O A, where A is the
           topmost element and B
           is the element below A.
        c. Push the result of evaluation
           on the stack
        [END OF IF]
Step 4: SET RESULT equal to the topmost element
        of the stack
Step 5: EXIT
```

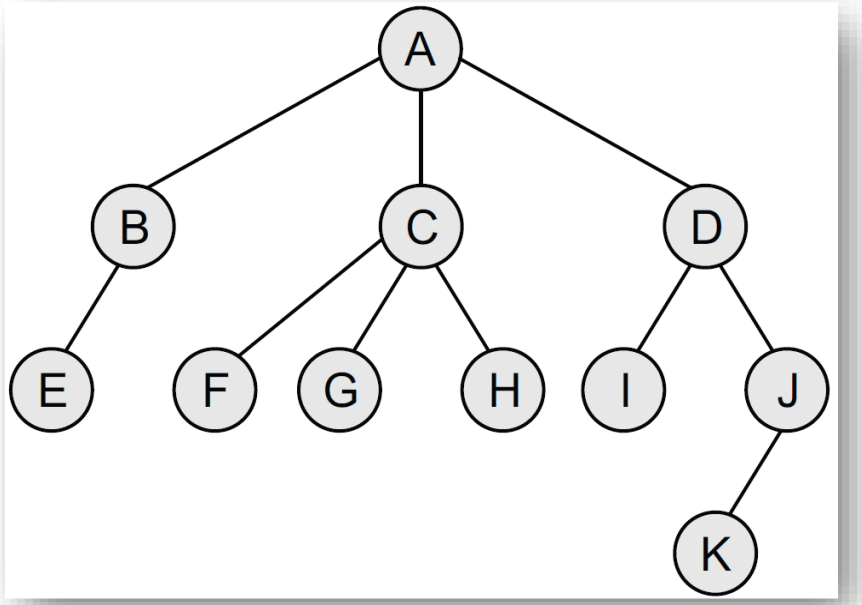
Infix: $12 \div 4$
Postfix: $12\ 4\ \div$

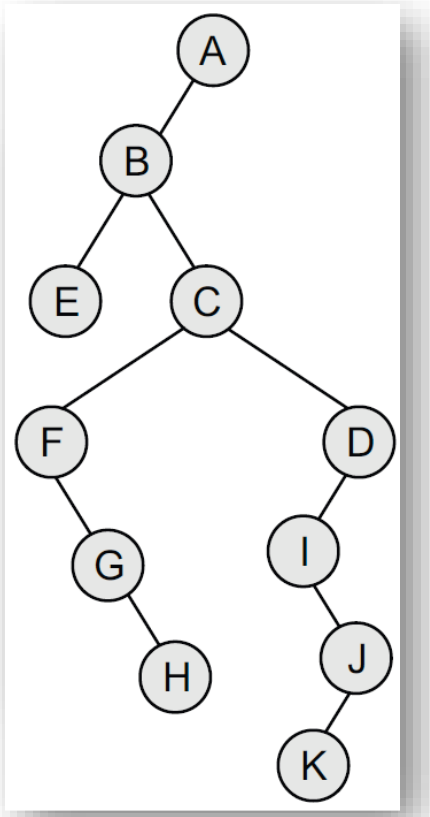
$A = 4$
$B = 12$

postfix	Stack	Top of the stack (Decimal)
6945&^|	0110	6
6945&^|	0110, 1001	9
6945&^|	0110, 1001, 0100	4
6945&^|	0110, 1001, 0100, 0101	5
6945&^|	0110, 1001, 0100	4
6945&^|	0110, 1101	13
6945&^|	1111	15

HW1-2 Directories

Description





A directory is given with the content (sub-directories or the files) inside. Please print the level-order traversal (from left to right) of the directory. Convert it to a binary tree and print the traversal (**prefix, postfix or infix**) specified at the end.

Test case A:
a b c d
b e
c f g h
d i j
j k
postfix

Equivalent to:
a/ b c d
b/ e
c/ f g h
d/ i j
j/ k

Output:
abcdefghijk
ehgfkjidcba

In this test case, *a* is the root directory, and

1. the order of the subdirectories / files listed shouldn't be changed, ensuring a unique binary tree
2. the directories aren't guaranteed to be given from top to bottom.

For example, an input like this is identical to the test case A.
Test case A':
b e
a b c d
d i j
j k
c f g h
postfix

Note:
1. Empty folders **are** in the test case.
For example,
a b c
b e
e
infix
2. Every name is unique, and of type ***char***.
3. The content of a directory will only be listed in one line. There is no test case like this:
a b
a c
b d
b f
prefix