Data Structures
Homework 3

基本題(80%)
1. Please show the processing of finding a S-G path through the given maze.
(a) Write down the sequence {1, 2, … } in which each empty cell is visited.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	S		1	1			1	1		1		1		1			1
1	1	1			1				1	1				1	1		1
1	1	1	1	1		1	1		1	1		1	1	1	1		1
1			1	1	1				1	1		1	1	1	1	1	1
1	1		1	1	1	1		1	1			1	1		1		1
1			1			1				1	1	1		1			1
1	1		1		1	1	1		1		1						1
1			1	1		1		1	1	1		1	1	1	1		1
1	1		1		1						1						1
1				1		1	1	1		1	1	1				1	1
1	1	1	1	1	1	1	1	1				1			G	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(b) Where in stack[i] are the following cells temporarily stored during the path finding process? Also show the 3-tuple $< x_i, y_i, dir >$. If the cells are pushed into the stack more than once, please show the last $<< x_i, y_i, dir >$ stored in the stack.

Cell	i (of stack [i])	$< x_i, y_i, dir >$
(2, 8)		
(4, 7)		
(7, 13)		
(9, 7)		
(10, 15)		

2. Use the algorithm presented in class (with a stack) to convert the following infix expressions into postfix expressions.

(a) 8 * ((4 + 2) / (5 - 3)) + 4 / (3 + (2 - 5))

(b) (a + (b - k)) * ((m / (n * p) + n) / (d + e))

3. Use the algorithm given in class (with one stack) to evaluate the following postfix expressions.

    1 2 * 3 4 5 6 7 + 8 9 - + - + - +

4. Convert the following postfix expressions into infix expressions.

    a b + c * d e - /

進階題(20%)

5. A prefix expression is similarly defined as a postfix expression except that the operator comes before the two operands that follow.

(a) Please show the pseudo code to convert an infix expression to a prefix expression by modifying the algorithm for converting an infix expression to a postfix expression.

(b) Please use the following example of infix expression to explain the steps of your pseudo code: (a+b)/(c*d)/e-f