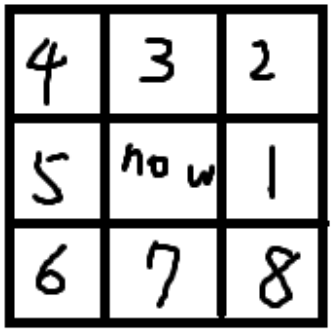
Data Structure HW3

41147009S 陳炫佑

P1

(A)

push order



maze visited order

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	S	1	1	1	4	5	1	1	9	1	26	1	25	1			1
1	1	1	2	3	1	6	7	8	1	1	27	23	24	1	1		1
1	1	1	1	1	15	1	1	10	1	1	22	1	1	1	1		1
1	72	71	1	1	1	14	11	12	1	1	21	1	1	1	1	1	1
1	1	70	1	1	1	1	13	1	1	28	20	1	1	80	1	37	1
1	73	69	1	63	62	1	16	17	18	19	1	1	1	79	1	36	1
1	1	68	1	61	1	1	1	57	1	29	1	31	32	33	34	35	1
1	74	67	1	1	60	1	56	1	1	1	30	1	1	1	1	38	1
1	1	66	1	64	1	58	59	55	54	52	53	1	42	41	39	40	1
1	75	76	65	1	77	1	1	1	51	1	1	1	43	44	45	1	1
1	1	1	1	1	1	1	1	1	78	50	49	48	1	46	47	G	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(B)

Cell	i	< xi, yi, dir >
(2, 8)	8	<2,8,5>
(4, 7)	10	<4,7,2>
(7, 13)	20	<7,13,5>
(9, 7)	37	<9,7,1>
(10, 15)	34	<10,15,5>

P2

(A)

842+53-/*4325-+/+

(B)

abk-+mnp*/n+de+/*

P3

26

step	string	note
1	1	
2	1 2	
3	1 2 *	
4	2	1*2=2
5	2 3	
6	2 3 4	
7	2 3 4 5	
8	2 3 4 5 6	
9	2 3 4 5 6 7	
10	2 3 4 5 6 7 +	
11	2 3 4 5 13	6+7=13
12	2 3 4 5 13 8	
13	2 3 4 5 13 8 9	
14	2 3 4 5 13 8 9 -	

step	string	note
15	2 3 4 5 13 17	8+9=17
16	2 3 4 5 13 17 +	
17	2 3 4 5 30	13+17=30
18	2 3 4 5 30 -	
19	2 3 4 -25	5-30=-25
20	2 3 4 -25 +	
21	2 3 -21	4+(-25)=-21
22	2 3 -21 -	
23	2 24	3-(-21)=24
24	2 24 +	
25	26	2+24=26

P4

```
(((a+b)*c)/(d-e))
```

P4

(A)

```
char input[10000];
int priority_stack[300];
stack<string> number;
stack<int> sign;
void fun(int now, int r) {
    if (r) {
        while (sign.top() != '(') {
            string b = number.top();
            number.pop();
            string a = number.top();
            number.pop();
            number.push(sign.top() + a + b);
            sign.pop();
        }
        sign.pop();
    } else {
        while (priority_stack[input[now]] <= priority_stack[sign.top()]) {
            string b = number.top();
            number.pop();
            string a = number.top();
            number.pop();
            number.push(sign.top() + a + b);
            sign.pop();
        }
    }
}
void reset() {
    priority_stack['('] = 0;
    priority_stack['+'] = 12;
    priority_stack['-'] = 12;
    priority_stack['*'] = 13;
    priority_stack['/'] = 13;
    priority_stack['%'] = 13;
    priority_stack[')'] = 19;
}
int main() {
    int i = 0;
    reset();
    scanf("%s", input);
    sign.push('(');
    for (i = 0; input[i] != 0x00; ++i) {
```

```
42          if ((input[i] <= '9' && input[i] >= 0) ||
43              (input[i] >= 'A' && input[i] <= 'Z') ||
44              (input[i] >= 'a' && input[i] <= 'z')) {
45              number.push(i);
46          } else if (input[i] == '(') {
47              sign.push(i);
48          } else if (input[i] == ')') {
49              fun(i, 1);
50          } else {
51              if (priority_stack[input[i]] <= priority_stack[sign.top()]) {
52                  fun(i, 0);
53              }
54              sign.push(i)
55          }
56      }
57      input[i] = ')';
58      fun(i, 1);
59      printf("%s", number.top());
60      return 0;
61  }
```

(B)

-/e/*cd+abf