

## 오픈랩 과제 #4 : Stacks & Queues

2020. 2학기

과제 : linked list로 구현된 stack을 이용하여 infix로 주어진 수식을 postfix로 변환한다.

(1) 입력 : 수식이 문자열로 주어진다. (expr.txt)

입력의 예

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

연산자는 +, -, \*, /, %만 포함한다.

피연산자는 single alphabet character이다.

(2) 자료구조와 알고리즘 (필요시 아래코드 수정 가능)

```
#include <stdio.h>
#define MAX_EXPR_SIZE 100
typedef enum {lparen, rparen, plus, minus, times, divide, mod, eos, operand}
    precedence;
precedence *stack;
char expr[MAX_EXPR_SIZE];
int isp[] = {0, 19, 12, 12, 13, 13, 13, 0}; // 교과서 3.6절 pp. 136 참고
int icp[] = {20, 19, 12, 12, 13, 13, 13, 0};

// 필요한 함수 선언
void postfix( );
precedence getToken(char*, int*);
void rExpr(FILE*, char*);
void main()
{
    파일을 오픈하고 수식을 읽어 배열 expr에 저장.
    rExpr(fp, expr);
    postfix( );
}
```

(3) 출력 : 변환된 postfix 수식을 출력한다.

a b c / - d e \* +

(4) 제출파일:

hw4\_학번.c

학번.docx (보고서: 코드 실행 결과 캡쳐 + 자신의 말로 코드/알고리즘 설명 한 것.  
두 가지 내용이 모두 있어야한다.)

제출기한: 10월 16일(금), 오후 5시 (늦은 제출은 받지 않음)