POSTFIX 계산기

OVERVIEW

- POSTFIX 형식으로 구성된 수식을 계산하여 결과를 도출한다

Condition

- POSTFIX로 구성된 수식을 계산한다
- 자료구조는 스택 링크드 리스트이다.

```
typedef struct  stackNode { // 스택의 노드를 구조체로 정의
    element data;
    struct stackNode* link;
} stackNode;

stackNode* top;
```

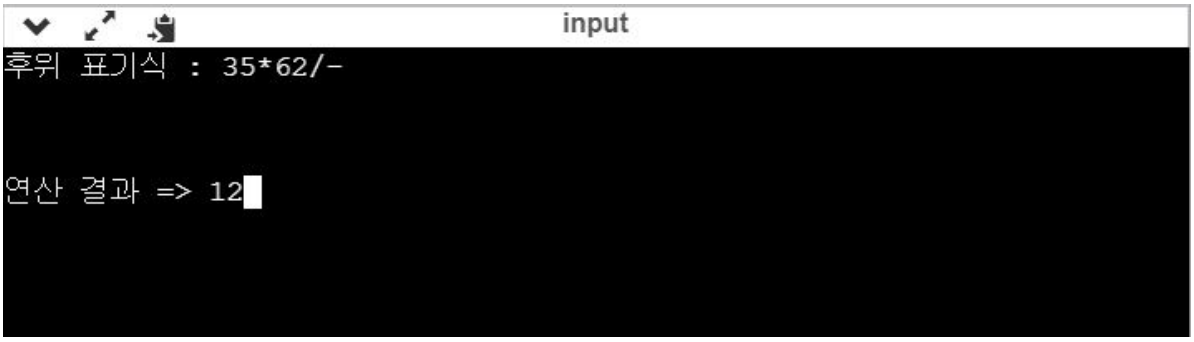
CODE

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef int element;        // 스택 원소(element)의 자료형을 int로 정의

typedef struct  stackNode { // 스택의 노드를 구조체로 정의
    element data;
    struct stackNode* link;
} stackNode;

stackNode* top;             // 스택의 top 노드를 지정하기 위해 포인터 top 선언

// 스택이 공백 상태인지 확인하는 연산
int isEmpty() {
    if (top == NULL) return 1;
    else return 0;
}

// 스택의 top에 원소를 삽입하는 연산
void push(element item) {
    stackNode* temp = (stackNode*)malloc(sizeof(stackNode));
    temp->data = item;
    temp->link = top;       // 삽입 노드를 top의 위에 연결
    top = temp;             // top 위치를 삽입 노드로 이동
}
```

```
// 스택의 top에서 원소를 삭제하는 연산
element pop() {
    element item;
    stackNode* temp = top;

    if (top == NULL) {      // 스택이 공백 리스트인 경우
        printf("\n\n Stack is empty !\n");
        return 0;
    }

    else {                  // 스택이 공백 리스트가 아닌 경우
        item = temp->data;
        top = temp->link;   // top 위치를 삭제 노드 아래로 이동
        free(temp);         // 삭제된 노드의 메모리 반환
        return item;        // 삭제된 원소 반환
    }
}

// 후위 표기법 수식을 계산하는 연산
element evalPostfix(char* exp) {
    int opr1, opr2, value, i = 0;
    // char형 포인터 매개변수로 받은 수식 exp의 길이를 계산하여 length 변수에 저장
    int length = strlen(exp);
    char symbol;
    top = NULL;
    for (int i = 0; i < length; i++) {
        switch (exp[i]) {
        case '+':
            opr2 = pop();   opr1 = pop();
            push(opr1 + opr2);  break;

        case '-':
            opr2 = pop();   opr1 = pop();
            push(opr1 - opr2);  break;

        case'*':
            opr2 = pop();   opr1 = pop();
            push(opr1 * opr2);  break;

        case'/':
            opr2 = pop();   opr1 = pop();
            push(opr1 / opr2);  break;

        default:
            value = exp[i] - '0';
            push(value);
            break;
        }
    }
    return pop();
}
// 수식 exp에 대한 처리를 마친 후 스택에 남아 있는 결과값을 pop하여 반환

void main(void) {
    int result;
    char* express = "35*62/-";
    printf("후위 표기식 : %s\n", express);
```

```
    result = evalPostfix(express);
    printf("\n\n연산 결과 => %d", result);

    getchar();
}
```

result



analysis

💡 구조체가 링크드 리스트로 구현되어 있어, is_full 을 확인할 필요 없이, PUSH 하면 새로운 구조체를 생성해낸다.

💡 동작 방식은 기호가 아닌 숫자일 때, 스택에 PUSH 하고, 기호를 만나면 스택에서 연산자와, 피연산자를 POP하고, 그 기호를 계산한 값을 다시 PUSH 하여 스택에 집어넣는다.

💡 계산이 모두 끝나면 스택에는 결과 값만 존재하므로, return pop을 하여 계산 결과를 리턴한다.

💡 주의할 점은, 스택에서 꺼낸 숫자가 차례대로 피연산자, 연산자 이므로, 반대로 계산을 해주어야 한다.