



# 5 주차 실습





## STACK 기본 소스

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef char element;
#define MAX_STACK_SIZE 100

typedef struct {
    element data[MAX_STACK_SIZE];
    int top;
} StackType;

void init_stack(StackType* s) {
    s->top = -1;
}

int is_empty(StackType* s) {
    return (s->top == -1);
}

int is_full(StackType* s) {
    return (s->top == (MAX_STACK_SIZE - 1));
}
```

```
void push(StackType* s, element item) {
    if (is_full(s)) {
        fprintf(stderr, "스택 포화 에러\n");
        return;
    }
    else s->data[++(s->top)] = item;
}

element pop(StackType* s) {
    if (is_empty(s)) {
        fprintf(stderr, "스택 공백 에러\n");
        exit(1);
    }
    else return s->data[(s->top)--];
}

element peek(StackType* s) {
    if (is_empty(s)) {
        fprintf(stderr, "스택 공백 에러\n");
        exit(1);
    }
    else return s->data[s->top];
}
```

# 실습 1 (partice\_9.c)



스택을 push pop을 이용하여 문자열 거꾸로 출력

- Kwangseob Kim 에 대한 문자열 배열을 정의 및 선언하고 이를 스택에 삽입 후 거꾸로 출력하시오

스택 소스 그대로 사용

```
int main(void) {  
    ???  
}
```

```
miK boesgnawK
```

# 실습 2 (partice\_10.c)

## 스택을 이용한 RLE(Run Length Encoding)

- 문자열 aaaabbbccdd 가 있다고 가정할 때, 이를 4a2b2c2d 로 압축
- 1. cnt(문자 숫자를 카운트하는 변수)를 1로 초기화
- 2. string[0] 값을 스택에 넣는다.
- 3. for 문을 돌면서[strlen(string)까지] 반복 문자를 카운트한다.
  - 1. peek를 통해 스택에 있는 문자를 char c에 저장
  - 2. for문 인덱스와 c 문자 값 확인
    - 같다면, cnt 값 하나 증가 / push
    - 다르다면, cnt 값 출력 및 문자 c 출력
  - 스택을 초기화 해주고 cnt 값도 1로 초기화 후 현재 문자를 push
- 4. for 문이 종료되면 마지막으로 cnt 값 출력 및 pop을 통해 마지막 문자 출력

스택 소스 그대로 사용

```
int main(void) {  
    StackType s;  
    char string[100];  
  
    init_stack(&s);  
    printf( " 문자열을 입력해주세요(소문자): " );  
    gets_s(string, 100);  
    ...  
}
```

# 실습 3 (partice\_11.c)



## 괄호 검사 - 수식 계산 (후위 표기식 계산/main)

```
int eval(char exp[]) {
    int op1, op2, value, i = 0;
    int len = strlen(exp);
    char ch;
    StackType s;
    init_stack(&s);
    for (i = 0; i < len; i++) {
        ch = exp[i];
        if (ch != '+' && ch != '-' && ch != '*' && ch != '/') {
            value = ch - '0';
            push(&s, value);
        }
        else {
            op2 = pop(&s);
            op1 = pop(&s);
            switch (ch) {
                case '+': push(&s, op1 + op2); break;
                case '-': push(&s, op1 - op2); break;
                case '*': push(&s, op1 * op2); break;
                case '/': push(&s, op1 / op2); break;
            }
        }
    }
    return pop(&s);
}
```

```
int main(void) {
    int result;
    printf("후위표기식은 82/3-32*+Wn");
    result = eval("82/3-32*+");
    printf("결과값은 %dWn", result);
    return 0;
}
```

# 실습 4 (partice\_12.c)



## 두 자리 수 후위 계산??

- partice\_9의 경우 한 자릿수 숫자에만 가능
- 2자리 이상 숫자에 대한 후위 계산을 하려면??
- $5 * (20 + (2 * 5 - 3))$  -> 5 20 2 5 \* 3 - + \*

1. 스택 데이터 자료형 char 에서 int 로 변경
2. eval 함수 안 for문 안에서 피연산자를 저장하는 부분을 ‘ ‘ 까지 체크하여 임시 문자열로 저장하고 atoi를 통해 정수로 변경

```
int main(void) {  
    int result;  
    printf("후위표기식은 2 200 2 5 * 3 - + *Wn");  
    result = eval("2 200 2 5 * 3 - + *");  
    printf("결과값은 %dWn", result);  
    return 0;  
}
```