연습문제 7, 8

- 다음은 어떤 수식의 후위 표기이다. 이 때 최초로 수행되는 연산은 어느 것인가?

  ABE+D*-

  1) B + E 2) E + A 3) D * B 4) B * E

- 크기가 5인 배열로 구현된 스택 A에 다음과 같이 삽입과 삭제가 되풀이 되었을 경우에 각 단계에서의 스택의 내용(1차원 배열의 내용, top의 값)을 나타내시오.

```
push(A, 1);
push(A, 2);
push(A, 3);
pop(A);
push(A, 4);
push(A, 5);
pop(A);
```

연습문제 9

- A와 B가 스택이라고 하고, a, b, c, d,가 객체라고 하고. 다음의 일련의 스택 연산을 수행한 뒤의 각각의 스택을 그려라.

```
push(A, a);
push(A, b);
push(A, c);
push(B, d);
push(B, pop(A));
push(A, pop(B));
pop(B);
```

스택 코드

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#define MAX_STACK_SIZE 100
typedef int element;
typedef struct {
    element data[MAX_STACK_SIZE];
    int top;
} StackType;// 스택 초기화 함수
void init_stack(StackType* s)
{
    s->top = -1;
}// 공백 상태 검출 함수
int is_empty(StackType* s)
{
    return (s->top == -1);
}
// 포화 상태 검출 함수
int is_full(StackType* s)
{
    return (s->top == (MAX_STACK_SIZE - 1));
}
// 삽입함수
```

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

연습문제 10

- 배열에 들어 있는 정수의 순서를 거꾸로 하는 프로그램을 작성해보자. 스택을 사용한다.
- 실행 결과
  - 정수 배열의 크기: 6
  - 정수를 입력하시오: 1 2 3 4 5 6
  - 반전된 정수 배열: 6 5 4 3 2 1

```c
int main(void)
{
    StackType s;
    int size;
    int array[MAX_STACK_SIZE];
    init_stack(&s);

    printf("정수 배열의 크기: ");
    scanf("%d", &size);

    printf("정수를 입력하시오: ");
    // 정수 입력 받는 코드 작성(반복문)
    // 스택에 순서대로 push

    printf("반전된 정수 배열: ");
    // 스택 내용 pop해서 출력 (반복문)

    printf("\n");
    return 0;
}
```

연습문제 11

- 수식에 있는 괄호의 번호를 출력하는 프로그램을 작성하라. 왼쪽 괄호가 나올 때마다 괄호 번호는 하나씩 증가한다. 오른쪽 괄호가 나오면 매칭되는 왼쪽 괄호 번호를 출력한다.
- 실행 결과 1
  - 수식: ((())(()))
  - 괄호 수: 1 2 3 3 2 4 5 5 4 1
- 실행 결과 2
  - 수식: (((((
  - 괄호 수: 1 2 3 4 5 5

```
int main(void)
{
    char line[100];
    printf("수식: ");
    gets_s(line, 100);
    printf("괄호수: ");
    check_matching(line);
    return 0;
}
```

- Hint
  - 슬라이드 내용 중 "괄호 검사 프로그램"의 check_matching함수 수정!!
    - 해당 case 절에 출력문 추가

연습문제 12

- 다음과 같이 문자열을 압축하는 프로그램을 작성하라. "4a3b"는 'a'가 4개, 'b'가 3개 있다는 의미이다. 이러한 압축 방법을 런길이(run length) 압축이라고 한다. 소문자와 대문자는 구별하지 않는다. 압축된 문자열에서는 소문자로 출력한다. 스택의 peek() 연산을 고려해보자.
- 실행 결과 1
  - 문자열을 입력하시오: aaaAbBb
  - 압축된 문자열: 4a3b
- 실행 결과 2
  - 문자열을 입력하시오: aaaAbBbaA
  - 압축된 문자열: 4a3b2a

```
void main() {
    int letter_count;
    char top;
    char line[100];
    StackType s;

    init_stack(&s);

    printf("문자열: ");
    gets_s(line, 100); // 사용자로부터 문자열 입력 받기

    for (int i = 0; line[i]; i++) { // 소문자로 변환 저장
        line[i] = tolower(line[i]);
    }

    printf("압축된 문자열: ");
    for (int i = 0; i < strlen(line); i++) {
        // 압축 문자열 계산 로직 작성
        // …
        // …
    }
}
```