

# C프로그래밍 (CSE2035) (실습19)



**Ji-Hwan Kim, Ph.D.**

**Dept. of Computer Science and Engineering**

**Sogang University**

**Seoul, Korea**

**Tel: +82-2-705-8924**

**Email : [kimjihwan@sogang.ac.kr](mailto:kimjihwan@sogang.ac.kr)**



## 제출 형식

---

1. 각 문제에 대한 소스 코드를 압축하여 사이버캠퍼스에 업로드
  - 압축 파일명: "[실습#]학번\_이름.zip" (#은 실습번호)
  - 각 소스코드 파일명: "cp실습번호\_학번\_p문제번호.c"
2. 업로드 후 제출 시, 제출 기한 내에 사이버캠퍼스에 제출
  - 화요일 실습의 경우: 이번주 수요일 오후 11시 59분까지
  - 목요일 실습의 경우: 이번주 금요일 오후 11시 59분까지
3. COPY 등의 문제 발생 시 실습 0점 및 각종 불이익을 줄 것



# Practice 1

- A[]의 숫자를 순서대로 **linked list**로 구현했던 실습 18 Practice 3번을 활용하여, **linked list**의 노드를 **head**와 가까운 순서대로 하나씩 삭제하는 프로그램을 작성한다.
- 원소가 하나씩 삭제될 때마다 **List**의 구조를 알 수 있게 원소를 모두 출력해 주어야 한다.
- 함수의 원형과 **main** 에서 입력이 들어가는 배열은 다음과 같다.

```
typedef struct node* nptr;
typedef struct node{
    int data;
    struct node *link;
}NODE;
// 원소를 하나씩 List에 넣는 함수
void Insert(NODE **, int);
// List 원소를 하나씩 제거하는 함수
void Delete(NODE**);
// List에 있는 원소를 모두 출력
void PrintAll(NODE*);
```

```
int A[10] = { 3, 9, 8, 2, 5, 10, 7, 1, 4, 6 };
```



# Practice 1

---

## ■ 제한 조건

- 입력은 Main 함수에서 미리 정해지며, 그 횟수만큼만 반복하면 된다. (10번)
- 전역변수는 사용하지 않는다.
- List의 원소를 제거할 때와 프로그램이 종료할 때 반드시 동적 할당을 한 변수/구조체들은 free를 해주어야 한다. (하지 않을 시 0점)



# Practice 1

- 프로그램 실행 결과

```
gr120160221@cspro:~/exercise/21$ ./21_1
3  9  8  2  5 10  7  1  4  6
9  8  2  5 10  7  1  4  6
8  2  5 10  7  1  4  6
2  5 10  7  1  4  6
5 10  7  1  4  6
10  7  1  4  6
7  1  4  6
1  4  6
4  6
6
Empty
```



## Practice 2

- **Linked list로 구현한 Stack을 이용해 Parentheses Matching 프로그램을 만든다.**
  - ‘(, ‘)’, ‘{, ‘}’, ‘[, ‘]’, ‘+’, ‘\*’ 과 정수로 이루어진 sequence의 괄호 symbol: ‘(, ‘)’, ‘{, ‘}’, ‘[, ‘]’ 의 짝이 맞게 이루어져 있는지를 확인한다.
  - Right parenthesis: ‘)’, ‘}’, ‘]’ 는 항상 앞에 짝이 맞는 left parenthesis가 존재해야 한다.
  - 짝이 맞는 parentheses를 balanced parentheses라고 한다.
- **Example of balanced parentheses**
  - $2 * 7$  : no parentheses 는 balanced parentheses
  - $(1+3)((2*16)+1)*(44+(17+9))$
- **Example of unbalanced parentheses**
  - $44 + 38)$  : a right parenthesis with no left parenthesis ‘(’
  - $(55+(12*11)$  : missing a right parenthesis ‘)’



## Practice 2

### ■ **Balanced parentheses**를 찾기 위한 방법

- 수업시간에 배운 Stack의 operation인 Push와 Pop을 이용한다.

### ■ **Hint**

- 빈 Stack을 만든다. (STACK\* pStack)
- Left parenthesis가 나올 때마다 Push operation을 수행
- Right parenthesis가 나올 때마다 Pop operation을 수행
- Parenthesis가 아닌 경우에는 operation을 수행하지 않는다.
- 더 이상 scan할 character가 없을 때 Stack이 비어있다면

“Matched Parentheses!”

- 마지막 character까지 scan을 마쳤을 때, Stack이 비어있지 않은 경우 / Stack은 비어있는데 Pop operation을 해야 하는 경우 중 하나이면

“Unmatched Parentheses!”

```
typedef struct node* nptr;

typedef struct {
    int count;
    nptr link;
}STACK;

typedef struct node{
    char data;
    nptr link;
}STACK_NODE;

void Push(STACK *, char);
int Pop(STACK *, char);
```



## Practice 2

### ■ Conditions

- Input으로 받는 수식은 100글자 이하라고 가정한다.
- Stack의 구현은 linked list를 이용한 구현 방법만 허용한다.
- page.4의 구조체 형식을 이용하여 구현 하도록 한다.
- gcc 컴파일러는 'bool' type선언이 불가하므로 Stack operation인 Push 또는 Pop의 성공 여부를 확인할 시에는 'int' type으로 선언한 함수와 변수를 사용한다.

### ■ Example of unmatched test

- $\{(a + b) - 3\} * 5 + [\cos(x + y) + 7] - 1 * 4$

### ■ Example of matched test

- $\{(a + b) - 3\} * 5 + [\{\cos(x + y) + 7\} - 1] * 4$





## Practice 2

---

- 프로그램 실행 결과

```
gr120160221@cspro:~/exercise/22$ ./22_2
(a+b)/3
Matched Parentheses!
gr120160221@cspro:~/exercise/22$ ./22_2
{(a+b)-2}*5+[cos(x+y)+2}-4]*4
Unmatched Parentheses!
```