# Spring 2019 Operating Systems
# Lab 03: Basic C Training III

## 0. 과제 수행 전 유의사항
프로그래밍은 자기 손으로 작성해야 한다. 아이디어를 서로 공유하고 도움을 받는 것은 좋으나, 다른 사람의 코드를 그대로 가져다 쓰는 것은 자신에게도 손해이고 다른 사람에게 피해를 끼치는 결과를 낳게되므로 허용하지 않는다. 과제 체크 시 두 사람의 코드가 유사하면, 두 사람 모두에게 과제 점수를 부여할 수 없으니 유의바란다.

## 1. 실습의 목표
포인터의 응용인 동적 할당을 사용하여 가변적인 테이블(배열)을 만드는 예제를 구현한다. 또한, 실제로 함수의 헤더파일에 필요한 것들만 선언하여 의존성을 만들고 Makefile로 그 의존성들을 확인하면서 자동 컴파일이 될 수 있도록 한다.

## 2. 작성해야할 파일
main.h, main.c, tables.h, tables.c, rows.h, rows.c, ops.c, Makefile

[main.h]

```
1
2 extern int** build_table(void);
3 extern void fill_table(int **);
4 extern void process_table(int **);
5
```

// main.h 에서 extern 이라는 키워드는 함수나 값이 외부(다른파일)에 있음을 알리는 것이다.

[main.c]

```
1
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <limits.h>
5
6 #include "main.h"
7
8 int
9 main(void)
10 {
11     int **table;
12
13     table = build_table();
14     fill_table(table);
15     process_table(table);
16
17     return 0;
18 }
19
20
```

[tables.h]

```
1
2 extern int row_minimum(int *);
3 extern int row_maximum(int *);
4 extern int row_average(int *);
5
```

[tables.c]

```
1
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 #include "tables.h"
6
7 int**
8 build_table(void)
9 {
```

```
10      int row_num, col_num;
11      int **table;
12      int row;
13
14      printf("\nEnter the number of rows in the table: ");
15      scanf("%d", &row_num);
16      table = (int **)calloc(row_num+1, sizeof(int *));
17      for (row = 0; row < row_num; row++) {
18          printf("Enter number of integers in row %d: ", row+1);
19          scanf("%d", &col_num);
20          table[row] = (int *)calloc(col_num+1, sizeof(int));
21          table[row][0] = col_num;
22      }
23      table[row] = NULL;
24
25      return table;
26  }
27
28  void
29  fill_table(int **table)
30  {
31      int row = 0;
32
33      printf("\n ==============================");
34      printf("\n Now we fill the table.\n");
35      printf("\n For each row enter the data");
36      printf("\n and press return: ");
37      printf("\n ==============================\n");
38
39      while (table[row] != NULL) {
40          printf("\n row %d (%d integers) ======> ", row+1, table[row][0]);
41          for (int column = 1; column <= *table[row]; column++) {
42              scanf("%d", table[row]+column);
43          }
44          row++;
45      }
46
47      return;
48  }
49
50  void
51  process_table(int **table)
52  {
53      int row = 0;
54      int row_min, row_max;
55      float row_ave;
56
57      while (table[row] != NULL) {
58          row_min = row_minimum(table[row]);
59          row_max = row_maximum(table[row]);
60          row_ave = row_average(table[row]);
61          printf("\nThe statistics for row %d ", row+1);
62          printf("\nThe minimum: %5d", row_min);
63          printf("\nThe maximum: %5d", row_max);
64          printf("\nThe average: %8.2f ", row_ave);
65          row++;
66      }
67      printf("\n");
68
69      return;
70  }
71
```

**[rows.h]**

```
1
2  extern int smaller(int, int);
3  extern int larger(int, int);
4
```

[rows.c]

```c

#include <stdio.h>
#include <stdlib.h>
#include <limits.h>

#include "rows.h"

int
row_minimum(int *row_ptr)
{
    int row_min = INT_MAX;

    for (int column = 1; column <= *row_ptr; column++) {
        row_min = smaller(row_min, *(row_ptr+column));
    }

    return row_min;
}

int
row_maximum(int *row_ptr)
{
    int row_max = INT_MIN;

    for (int column = 1; column <= *row_ptr; column++) {
        row_max = larger(row_max, *(row_ptr+column));
    }

    return row_max;
}

int
row_average(int *row_ptr)
{
    float total = 0;
    float row_ave;

    for (int column = 1; column <= *row_ptr; column++) {
        total += (float)*(row_ptr+column);
    }
    row_ave = total / *row_ptr;

    return row_ave;
}

```

[ops.c]

```c

#include <stdio.h>

int
smaller(int first, int second)
{
    return (first < second ? first : second);
}

int
larger(int first, int second)
{
    return (first > second ? first : second);
}

```

### 3. 컴파일
6개의 .c, .h 파일을 작성하였다면, 이번에는 작성한 파일들을 make 유틸리티를 통해 컴파일을 한다. 따라서, Makefile을 스스로 작성하고 make 명령을 통해 컴파일 해야한다.

Makefile은 PPT의 내용을 보고 스스로 작성한다. 아래와 같은 출력을 보여야 한다.

```
kwp@A1409-OSLAB-01:~/.toy/lab3/task$ make
gcc -c main.c
gcc -c tables.c
gcc -c rows.c
gcc -c ops.c
gcc -o lab03_kwp main.o tables.o rows.o ops.o
```

### 4. 출력내용 확인
본 과제는 입력을 받는 과제이다. 입력은 배열의 크기를 3이상으로 자유롭게 한다. 실행 예는 다음과 같다.

```
kwp@A1409-OSLAB-01:~/.toy/lab3/task$ ./lab03_kwp

Enter the number of rows in the table: 3
Enter number of integers in row 1: 3
Enter number of integers in row 2: 3
Enter number of integers in row 3: 3

 ============================
 Now we fill the table.

 For each row enter the data
 and press return:
 ============================

 row 1 (3 integers) =====> 1
2
3

 row 2 (3 integers) =====> 4
5
6

 row 3 (3 integers) =====> 7
8
9

The statistics for row 1
The minimum:     1
The maximum:     3
The average:     2.00
The statistics for row 2
The minimum:     4
The maximum:     6
The average:     5.00
The statistics for row 3
The minimum:     7
The maximum:     9
The average:     8.00
```

### 5. 그래프 작성
메인 함수를 시작으로 각 함수의 호출 과정을 그래프로 작성한다. 그래프는 손으로 작성해도 된다. 다만, 캡쳐하여 첨부한다.

### 6. 제출
위 파일을 코딩하고 테스트 한 다음, ls를 입력하여 자신의 파일 목록을 캡쳐한 후 추가하고 실행결과와 주석이 달린 코드를 워드 파일로 추합하여 제출한다.

코드로 적혀있는 내용을 단순히 한글로 옮긴 주석에는 점수를 부여할 수 없으니 참고하기 바란다.

자신의 디렉토리 목록과 실행과정은 스크린샷을 이용하도록 하고, 소스코드는 자신의 것을 Copy하여 첨부한다.

모든 내용은 MS워드나 Pages를 이용하여 한 개의 파일로 작성한다.