

## PART 09 ► 프로그래밍 실습 해답

1. [프로그램 9-9]를 다음과 같이 확장하라.

- ① 입력으로 각 학생의 국, 영, 수 점수를 입력받는다.
- ② 각 학생의 점수의 합과 각 과목의 점수의 합을 계산하여 출력한다.

```
#include <stdio.h>

int main()
{
    int score[5][3];
    int stud[5], sum;
    int kor = 0, eng = 0, math = 0;

    for(int i=0;i<5;i++) {
        printf("%d번째 학생의 국어, 영어, 수학 점수 입력: ", i+1);
        for(int j=0;j<3;j++) {
            scanf("%d", &score[i][j]);
        }
    }

    for(int i=0;i<5;i++) {
        sum = 0;
        for(int j=0;j<3;j++) {
            sum += score[i][j];
        }
        kor += score[i][0];
        eng += score[i][1];
        math += score[i][2];
        stud[i] = sum;
    }

    for(int i=0;i<5;i++)
        printf("%d번째 학생의 점수 합: %d \n", i+1, stud[i]);

    printf("국어 점수의 합: %d, 영어 점수의 합: %d, 수학 점수의 합: %d\n",
kor, eng, math);

    return 0;
}
```

2. 틱택토(Tick-Tac-Toe) 프로그램을 수행하는 프로그램을 작성하라. 틱택토 프로그램이란 3×3 바둑판 모양의 말판에 두 사람이 번갈아 가며 돌을 놓은 다음, 가로나 세로, 대각선으로 세 개를 같은 돌로 채우면 이기는 경기이다.

(1) 틱택토 말판은 다음과 같이 출력하자.

```

      1   2   3
    +---+---+---+
A  |   |   |   |
    +---+---+---+
B  |   |   |   |
    +---+---+---+
C  |   |   |   |
    +---+---+---+

```

- (2) 틱택토 프로그램은 각 위치에 대한 번호를 입력하면 해당 위치에 돌을 놓을 수 있는지 판단하고, 놓을 수 있는 경우에는 해당 돌을 놓고 변화된 말판을 출력한다. 각 돌은 O, X로 출력하자. 예컨대 O 돌이 먼저 두는 사람이고 b3라고 입력했을 경우 다음과 같이 말판을 출력해야 한다.

```

      1   2   3
    +---+---+---+
A  |   |   |   |
    +---+---+---+
B  |   |   | X |
    +---+---+---+
C  |   |   |   |
    +---+---+---+

```

- (3) 매번 돌을 놓을 때마다 어느 돌이 이겼는지 판단하고 특정한 사람이 이겼을 경우에는 축하한다는 메시지를 출력한다.

```

//*****
//  tictactoe.c
//
//  틱택토 게임을 할 수 있게 해 주는 프로그램
//*****

#include <stdio.h>
#include <assert.h>
#include <ctype.h>

```

```

enum Turn {NONE = 0, A = 1, B = 10, DRAW = -1};

void init_board(int board[3][3]);
void show_board(int board[3][3]);
enum Turn winner(int board[3][3]);
enum Turn flip(enum Turn player);
void tic(int board[3][3], enum Turn *player);
void congratulation(enum Turn who);

int main()
{
    enum Turn player = A;
    enum Turn who = NONE;
    int b[3][3];

    init_board(b);
    show_board(b);
    while ((who = winner(b)) == NONE && who != DRAW) {
        tic(b, &player);
        show_board(b);
    }
    congratulation(who);

    return 0;
}

void init_board(int board[3][3])
{
    int i = 0, j = 0;
    for (i = 0; i < 3; ++i)
        for (j = 0; j < 3; ++j)
            board[i][j] = NONE;
}

char pebble(enum Turn pos)
{
    switch (pos) {
        case 0: return ' ';
        case 1: return 'O';
        case 10: return 'X';
        default: assert(0);
    }
}

void show_board(int board[3][3])
{

```

```

    int i = 0, j = 0;
    printf("      1   2   3\n");
    printf("    +---+---+---+\n");
    for (i = 0; i < 3; ++i) {
        printf("  %c | ", 'A' + i);
        for (j = 0; j < 3; ++j) {
            printf("%c | ", pebble(board[i][j]));
        }
        printf("\n    +---+---+---+\n");
    }
}

enum Turn winner(int board[3][3])
{
    int sum[8] = {0};
    int i = 0, j = 0;
    for (i = 0; i < 3; ++i) {
        for (j = 0; j < 3; ++j) {
            sum[i] += board[i][j];
            sum[3+i] += board[j][i];
        }
        sum[6] += board[i][i];
        sum[7] += board[i][2-i];
    }
    // 승자를 찾음
    for (i = 0; i < 8; ++i) {
        if (sum[i] == A * 3) return A;
        if (sum[i] == B * 3) return B;
    }
    // 게임이 진행중인가 판별
    for (i = 0; i < 8; ++i) {
        if (sum[i] == A * 2 || sum[i] == A * 1) return NONE;
        if (sum[i] == B * 2 || sum[i] == B * 1) return NONE;
        if (sum[i] == 0) return NONE;
    }
    return DRAW;
}

enum Turn flip(enum Turn player)
{
    return (player == A)? B: A;
}

char player_name(enum Turn player)
{
    return (player == A)? 'A': 'B';
}

```

```

}

char pebble_shape(enum Turn player)
{
    return (player == A)? 'O': 'X';
}

int is_wrong_place(int board[3][3], int row, int col)
{
    return ((row < 0 || row > 2) || (col < 0 || col > 2))? 1:
           (board[row][col] != NONE);
}

void tic(int board[3][3], enum Turn *pp)
{
    char row, col;
    enum Turn player = *pp;
    do {
        char buf[80];
        printf("선수 %c: 어디에 두시겠습니까(예: b3) ? ",
               player_name(player));
        scanf("%c%c", &row, &col);
        gets(buf);
        row -= (isupper(row))? 'A' : 'a';
        col -= '1';
    } while (is_wrong_place(board, row, col)
            && printf("Wrong position !\n"));
    board[row][col] = player;
    *pp = flip(player);
}

void congratulation(enum Turn player)
{
    switch (player) {
        case A:
        case B:
            printf("\n");
            printf("*****\n");
            printf("*** 축하합니다! 선수 %c(%c) 승리! ***\n",
                   player_name(player), pebble_shape(player));
            printf("*****\n");
            printf("\n");
            break;
        default:
            printf("\n");
            printf("*****\n");
    }
}

```

```
        printf("***    아쉽네요. 무승부입니다.    ***\n");  
        printf("*****\n");  
        printf("\n");  
        break;  
    }  
}
```