

PART 17-2. 구조체 활용, 공용체, 열거형

◆ 연습문제 ◆

1. 다음과 같이 구조체 변수와 구조체 포인터를 선언했을 때, mp를 사용하여 m1에 저장된 값을 출력하세요. 멤버를 참조할 때는 -> 연산자를 사용합니다.

```
struct marriage {  
    char name[20];  
    int age;  
    char sex;  
    double height;  
};
```

```
struct marriage m1;  
struct marriage *mp = &m1;
```

정답

```
printf("이름 : %s", mp -> name);  
printf("나이 : %d", mp -> age);  
printf("성별 : %c", mp -> sex);  
printf("키 : %.1lf", mp -> height);
```

해설 : 없음

2. 다음 코드와 같이 연결 리스트를 만들 수 있도록 코드의 빈 칸을 채우세요.

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

```

typedef struct train Train;
struct train {
    int seats;
    (      ) *next;
};

int main(void) {
    Train *head = NULL, *tail;

    int i;

    for (i = 0; i < 5; i++) {
        if (head == NULL) {
            head = tail = (Train *) malloc(sizeof(Train));
        } else {
            tail -> next = (Train *) malloc(sizeof(Train));
            (
                );
        }
    }

    return 0;
}

```

정답 : ① Train 또는 struct train ② tail = tail -> next;

해설 : 없음

3. 다음 프로그램의 실행 결과를 예상해 보세요.

```

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

typedef enum { CYAN, MAGENTA, YELLOW = 5, BLACK } COLOR;
typedef enum { UP, DOWN, LEFT, RIGHT } ARROW;

```

```

int main(void) {
    COLOR my_color = YELLOW, c;
    ARROW direction = UP;

    for (c = CYAN; c <= BLACK; c++) {
        direction++;
        direction = direction % 4;
        if (c == my_color) {
            break;
        }
    }

    switch(direction) {
        case UP:
            printf("현재 방향 : 위");
            break;
        case DOWN:
            printf("현재 방향 : 아래");
            break;
        case LEFT:
            printf("현재 방향 : 왼쪽");
            break;
        case RIGHT:
            printf("현재 방향 : 오른쪽");
            break;
    }

    return 0;
}

```

정답 : 왼쪽

해설 : 없음

4. 비밀번호를 입력하고 출력하는 프로그램을 작성하세요. 단, 공용체와 열거형을 사용하여 비밀번호의 입력 형식을 숫자와 전화번호 중에 선택할 수 있도록 합니다.

```
enum = { NUMBER = 1, PHONE_NUMBER};
```

```
typedef union {  
    int num;  
    char phone[20];  
} Password;
```

[실행 결과 1]

1. 숫자 4자리
 2. 전화번호
- 원하는 비밀번호 입력형식 : 1
비밀번호 입력 : 1234
입력한 비밀번호 : 1234

[실행 결과 2]

1. 숫자 4자리
 2. 전화번호
- 원하는 비밀번호 입력형식 : 2
비밀번호 입력 : 01012345678
입력한 비밀번호 : 01012345678

정답

```
#include <stdio.h>
```

```
enum { NUMBER = 1, PHONE_NUMBER };
```

```

typedef union {
    int num;
    char phone[20];
} Password;

int main(void) {
    Password my_pw;
    int sel;

    printf("1. 숫자 4자리\n");
    printf("2. 전화번호\n");
    printf("원하는 비밀번호 입력방식 : ");
    scanf("%d", &sel);

    printf("비밀번호 입력 : ");

    switch (sel) {
        case NUMBER:
            scanf("%d", &my_pw.num);
            break;
        case PHONE_NUMBER:
            scanf("%s", my_pw.phone);
            break;
    }

    printf("입력한 비밀번호 : ");

    switch (sel) {
        case NUMBER:
            scanf("%d", &my_pw.num);
            break;
        case PHONE_NUMBER:
            scanf("%s", my_pw.phone);
            break;
    }

```

```

    }

    return 0;
}

```

해설 : 없음

5. 신입사원의 이름, 학점, 영어점수를 입력하여 엘리트 사원의 데이터를 출력하는 프로그램을 작성하세요. 단, 다음 조건에 맞도록 작성합니다.

- 신입사원의 수는 5명으로 하며 구조체 배열을 선언한다.
- 구조체 배열에는 키보드로 데이터를 입력한다.
- 엘리트 사원의 조건은 학점이 4.0 이상이고 영어점수가 900점 이상이다.
- 구조체의 선언과 main 함수는 다음과 같이 작성한다.

```

struct profile {
    char name[20];
    double grade;
    int english;
};

int main(void) {
    struct profile new_staff[5];

    input_data(new_staff);
    elite(new_staff);

    return 0;
}

```

실행 결과

이름, 학점, 영어 점수를 입력하세요.
홍길동 3.7 840

이순신 4.1 920
장보고 3.9 940
유관순 4.3 750
세종대왕 4.5 990
이순신, 4.1, 920
세종대왕, 4.5, 990

정답

```
#include <stdio.h>
```

```
struct profile {  
    char name[20];  
    double grade;  
    int english;  
};
```

```
void input_data(struct profile *ps);  
void elite(struct profile *ps);
```

```
int main(void) {  
    struct profile new_staff[5];  
  
    input_data(new_staff);  
    elite(new_staff);  
  
    return 0;  
}
```

```
void input_data(struct profile *ps) {  
    int i;  
  
    printf("이름, 학점, 영어점수를 입력하세요.\n");
```

```

for (i = 0; i < 5; i++) {
    scanf("%s%1f%d", ( ps + i ) -> name, &(( ps + i ) -> grade),
        &(( ps + i ) -> english));
}
}

void elite(struct profile *ps) {
    int i;

    for (i = 0; i < 5; i++) {
        if ((( ( ps + i ) -> grade ) >= 4.0) && ((( ps + i ) -> english )
            >= 900)) {
            printf("%s, %.1lf, %dWn",
                ( ps + i ) -> name, ( ps + i ) -> grade, ( ps + i ) ->
                english);
        }
    }
}

```

해설 : 없음