

CHAPTER 6

구조체의 이해: 구조체
배열, 중첩 구조체 실습

실습 1

문제 1)

아래의 구조체를 사용하여 다음과 같이 출력하시오

```
struct student
{
    int num;
    char name[20];
    int score[5];
    double avg;
};
```

```
학번 : 315
이름 : 홍길동
점수 : 82 93 74 90 65
평균 : 80.8
```

실습 1 정답

```
1 #include <stdio.h>
2
3 struct student                // 학생 구조체
4 {
5     int num;                  // 학번
6     char name[20];            // 이름
7     int score[5];             // 점수 배열
8     double avg;               // 평균
9 };
10 struct student print_avg(struct student s);
11
12 int main(void)
13 {
14     struct student s1        // 구조체 변수의 선언과 초기화
15     = { 315, "홍길동", {82, 93, 74, 90, 65} };
16
17     s1 = print_avg(s1);       // 구조체 변수 전달, 반환값 저장
18     printf("평균 : %.1lf\n", s1.avg); // 함수가 계산한 평균 출력
19
20     return 0;
21 }
22 // 구조체 변수의 데이터를 출력하고 평균 계산
23 struct student print_avg(struct student s) // 매개변수와 반환형은 모두 구조체
24 {
25     int i, sum = 0;
26
27     printf("학번 : %d\n", s.num);
28     printf("이름 : %s\n", s.name);
29     printf("점수 : ");
30
31     for (i = 0; i < 5; i++)
32     {
33         printf("%5d", s.score[i]); // 배열 멤버의 값 출력
34         sum += s.score[i];          // 총점 계산
35     }
36     printf("\n");
37     s.avg = sum / 5.0;              // 평균 계산
38
39     return s;                      // 평균이 계산된 구조체 변수 반환
40 }
```

```
학번 : 315
이름 : 홍길동
점수 :      82      93      74      90      65
평균 : 80.8
```

실습 2

문제 2)

구조체를 사용해서 사람의 신체 정보를 입력 받고 출력하기

```
대상자의 정보를 입력하세요
```

```
이름 : 홍길동
```

```
나이 : 25
```

```
키 : 170
```

```
몸무게 : 60
```

```
입력한 정보는 다음과 같습니다.
```

```
홍길동 : 25세, 170.0cm, 60.0kg
```

실습 2 정답

```
1  #include <stdio.h>
2
3  typedef struct People
4  {
5      char name[12];
6      unsigned short int age;
7      float height;
8      float weight;
9  } Person;
10
11 void main()
12 {
13     Person data;
14
15     printf("대상자의 정보를 입력하세요\n");
16     printf("이름 : ");
17     scanf("%s", data.name);
18     printf("나이 : ");
19     /* %u라고 적으면 unsigned int 형으로 처리되기 때문에
20      * unsigned short int 형으로 입력 받으려면 %hu 사용 */
21     scanf("%hu", &data.age);
22     printf("키 : ");
23     scanf("%f", &data.height);
24     printf("몸무게 : ");
25     scanf("%f", &data.weight);
26
27     printf("\n입력한 정보는 다음과 같습니다.\n");
28     /* 키와 몸무게는 소수점 첫째자리까지만 보여주기 위해서
29      * %.1f 형식으로 출력함 */
30     printf("%s : %d세, %.1fcm, %.1fkg\n", data.name, data.age, data.height, data.weight);
31 }
```

대상자의 정보를 입력하세요

이름 : 홍길동

나이 : 25

키 : 170

몸무게 : 60

입력한 정보는 다음과 같습니다.

홍길동 : 25세, 170.0cm, 60.0kg

실습 3

문제 3)

배열을 사용하여 친구 정보 관리하기
등록한 친구 수는 최대 5명으로 설정

```
[ 메뉴 ]
=====
1. 친구 추가
2. 친구 목록 보기
3. 종료
=====
번호 선택 : 1

새로운 친구 정보를 입력하세요
1. 이름 : 홍길동
2. 나이 : 24
3. 키 : 170
4. 몸무게 : 59
입력을 완료했습니다.

[ 메뉴 ]
=====
1. 친구 추가
2. 친구 목록 보기
3. 종료
=====
번호 선택 : 2

등록된 친구 목록
=====
홍길동      , 24, 170.00, 59.00
=====
```

실습 3 정답 (1/3)

```
1 #include <stdio.h>
2 #define MAX_COUNT 5 /* 친구 등록 가능한 최대 수 */
3
4 typedef char NAME_TYPE[14];
5
6 /* 친구를 추가하는 함수 : 성공하면 1을 반환하고 실패하면 0을 반환 함
7  main 함수에서 선언된 name, age, height, weight 배열의 시작 주소를 포인터로
8  받아서 사용. count는 현재까지 입력된 친구의 수를 기억하는 변수임 */
9 int AddFriend(NAME_TYPE *p_name, unsigned short int *p_age,
10              float *p_height, float *p_weight, int count)
11 {
12     if(count < MAX_COUNT) { /* 입력 가능한 최대 수를 넘었는지 체크 */
13         printf("\n새로운 친구 정보를 입력하세요\n");
14         printf("1. 이름 : ");
15         scanf("%s", *(p_name + count)); /* name 배열의 count 위치에 이름 입력 */
16         printf("2. 나이 : ");
17         scanf("%hu", p_age + count); /* age 배열의 count 위치에 나이 입력 */
18         printf("3. 키 : ");
19         scanf("%f", p_height + count); /* height 배열의 count 위치에 키 입력 */
20         printf("4. 몸무게 : ");
21         scanf("%f", p_weight + count); /* weight 배열의 count 위치에 몸무게 입력 */
22         printf("입력을 완료했습니다. \n\n");
23         return 1; /* 친구 추가 성공 */
24     } else {
25         /* 입력 가능한 최대 수를 넘었을 때 오류 상태 출력 */
26         printf("최대 인원을 초과하여 입력을 할 수 없습니다. \n");
27         printf("최대 %d명까지만 관리 가능합니다. \n\n", MAX_COUNT);
28     }
29     return 0; /* 친구 추가 실패 */
30 }
31 void ShowFriendList(NAME_TYPE *p_name, unsigned short int *p_age,
32                    float *p_height, float *p_weight, int count)
33 {
34     int i;
35     if(count > 0) { /* 등록된 친구가 있으면 그 수만큼 반복하면서 친구 정보 출력 */
36         printf("\n등록된 친구 목록\n");
37         printf("=====\n");
38         for(i = 0; i < count; i++) {
39             printf("%-14s, %3d, %6.2f, %6.2f\n", *(p_name + i), *(p_age + i),
40                *(p_height + i), *(p_weight + i));
41         }
42         printf("=====\n\n");
43     } else { /* 등록된 친구가 없으면 오류 상태 출력 */
44         printf("\n등록된 친구가 없습니다.\n\n");
45     }
46 }
```

실습 3 정답 (2/3)

```
47 void main()
48 {
49     NAME_TYPE name[MAX_COUNT]; /* 친구 이름을 저장할 배열 */
50     unsigned short int age[MAX_COUNT]; /* 친구 나이를 저장할 배열 */
51     float height[MAX_COUNT]; /* 친구 키를 저장할 배열 */
52     float weight[MAX_COUNT]; /* 친구 몸무게를 저장할 배열 */
53     int count = 0, num; /* count : 등록된 친구 수 */
54
55     while(1) { /* 무한 루프 : 사용자가 3을 누르면 break 문으로 종료 시킴 */
56         /* 메뉴를 화면에 출력 */
57         printf("    [ 메뉴 ]    \n");
58         printf("===== \n");
59         printf("1. 친구 추가      \n");
60         printf("2. 친구 목록 보기 \n");
61         printf("3. 종료          \n");
62         printf("===== \n");
63         printf("번호 선택 : ");
64         scanf("%d", &num); /* 사용자에게 번호를 입력 받음 */
65
66         if(num == 1) { /* 1번 : 친구 추가를 선택한 경우 */
67             /* AddFriend 함수가 1을 반환하면 정상적으로 추가된 경우
68              그래서 1을 반환했을 때만 등록된 친구 수 증가시킴
69              이름, 나이, 키, 몸무게 배열의 주소를 인자로 넘겨 줌 */
70             if(1 == AddFriend(name, age, height, weight, count)) count++;
71         } else if(num == 2) { /* 2번 : 친구 목록 보기를 선택한 경우 */
72             /* 이름, 나이, 키, 몸무게 배열의 주소를 인자로 넘겨 줌 */
73             ShowFriendList(name, age, height, weight, count);
74         } else if(num == 3){ /* 3번 : 반복문을 빠져나가 종료 함 */
75             break;
76         } else {
77             /* 번호가 유효하지 않는 경우에 오류 출력 */
78             printf("1~3 번호만 선택할 수 있습니다!!\n\n");
79         }
80     }
81 }
```


실습 3 정답 (3/3)

```
[ 메뉴 ]
=====
1. 친구 추가
2. 친구 목록 보기
3. 종료
=====
번호 선택 : 1

새로운 친구 정보를 입력하세요
1. 이름 : 홍길동
2. 나이 : 24
3. 키 : 170
4. 몸무게 : 59
입력을 완료했습니다.
```

```
[ 메뉴 ]
=====
1. 친구 추가
2. 친구 목록 보기
3. 종료
=====
번호 선택 : 2

등록된 친구 목록
=====
홍길동      ,  24, 170.00,  59.00
=====
```

실습 4

문제 4)

구조체를 사용하여 친구 정보 관리하기
등록한 친구 수는 최대 5명으로 설정

```
[ 메뉴 ]
=====
1. 친구 추가
2. 친구 목록 보기
3. 종료
=====
번호 선택 : 1

새로운 친구 정보를 입력하세요
1. 이름 : 홍길동
2. 나이 : 24
3. 키 : 170
4. 몸무게 : 59
입력을 완료했습니다.

[ 메뉴 ]
=====
1. 친구 추가
2. 친구 목록 보기
3. 종료
=====
번호 선택 : 2

등록된 친구 목록
=====
홍길동      ,  24, 170.00,  59.00
=====
```

실습 4 정답 (1/3)

```
1  #include <stdio.h>
2  #define MAX_COUNT      5  /* 친구 등록 가능한 최대 수 */
3
4  typedef struct People
5  {
6      char name[14];          /* 이름 */
7      unsigned short int age;  /* 나이 */
8      float height;           /* 키 */
9      float weight;           /* 몸무게 */
10 } Person;
11
12 int AddFriend(Person *p_friend, int count)
13 {
14     if(count < MAX_COUNT) { /* 입력 가능한 최대 수를 넘었는지 체크 */
15         p_friend = p_friend + count; /* friends 배열의 count 위치로 주소 이동 */
16         printf("\n새로운 친구 정보를 입력하세요\n");
17         printf("1. 이름 : ");
18         scanf("%s", p_friend->name); /* 구조체의 name 항목에 입력 */
19         printf("2. 나이 : ");
20         scanf("%hu", &p_friend->age); /* 구조체의 age 항목에 입력 */
21         printf("3. 키 : ");
22         scanf("%f", &p_friend->height); /* 구조체의 height 항목에 입력 */
23         printf("4. 몸무게 : ");
24         scanf("%f", &p_friend->weight); /* 구조체의 weight 항목에 입력 */
25         printf("입력을 완료했습니다. \n\n");
26         return 1; /* 친구 추가 성공 */
27     } else {
28         /* 입력 가능한 최대 수를 넘었을 때 오류 상태 출력 */
29         printf("최대 인원을 초과하여 입력을 할 수 없습니다. \n");
30         printf("최대 %d명까지만 관리 가능합니다. \n\n", MAX_COUNT);
31     }
32     return 0; /* 친구 추가 실패 */
33 }
```

실습 4 정답 (2/3)

```
34 void ShowFriendList(Person *p_friend, int count)
35 {
36     int i;
37     if(count > 0) { /* 등록된 친구가 있으면 그 수만큼 반복하면서 친구 정보 출력 */
38         printf("\n등록된 친구 목록\n");
39         printf("=====\n");
40         for(i = 0; i < count; i++) {
41             printf("%-14s, %3d, %6.2f, %6.2f\n", p_friend->name, p_friend->age,
42                 p_friend->height, p_friend->weight);
43             p_friend++; /* 다음 위치에 있는 친구 정보로 주소 이동 */
44         }
45         printf("=====\n\n");
46     } else { /* 등록된 친구가 없으면 오류 상태 출력 */
47         printf("\n등록된 친구가 없습니다.\n\n");
48     }
49 }
50 void main()
51 {
52     Person friends[MAX_COUNT]; /* 친구 정보를 저장할 배열 */
53     int count = 0, num; /* count : 등록된 친구 수 */
54
55     while(1) { /* 무한 루프 : 사용자가 3을 누르면 break 문으로 종료 시킴 */
56         /* 메뉴를 화면에 출력 */
57         printf("    [ 메뉴 ]    \n");
58         printf("=====\n");
59         printf("1. 친구 추가    \n");
60         printf("2. 친구 목록 보기 \n");
61         printf("3. 종료        \n");
62         printf("=====\n");
63         printf("번호 선택 : ");
64         scanf("%d", &num); /* 사용자에게 번호를 입력 받음 */
65
66         if(num == 1) { /* 1번 : 친구 추가를 선택한 경우 */
67             if(1 == AddFriend(friends, count)) count++;
68         } else if(num == 2) { /* 2번 : 친구 목록 보기를 선택한 경우 */
69             /* friends 배열의 주소를 인자로 넘겨 줌 */
70             ShowFriendList(friends, count);
71         } else if(num == 3){ /* 3번 : 반복문을 빠져나가 종료 함 */
72             break;
73         } else {
74             /* 번호가 유효하지 않는 경우에 오류 출력 */
75             printf("1~3 번호만 선택할 수 있습니다!!\n\n");
76         }
77     }
78 }
```

실습 4 정답 (3/3)

```
[ 메뉴 ]
=====
1. 친구 추가
2. 친구 목록 보기
3. 종료
=====
번호 선택 : 1

새로운 친구 정보를 입력하세요
1. 이름 : 홍길동
2. 나이 : 24
3. 키 : 170
4. 몸무게 : 59
입력을 완료했습니다.

[ 메뉴 ]
=====
1. 친구 추가
2. 친구 목록 보기
3. 종료
=====
번호 선택 : 2

등록된 친구 목록
=====
홍길동      , 24, 170.00, 59.00
=====
```

실습 5

문제 5)

문자열에 저장된 시간을 읽어오기(strtok(), atoi(), mktime() 함수)
정수값으로 변환하여 출력하시오

```
char strTime[] = "2006/05/15 09:05:07";
```

```
char* token;  
struct tm t;  
time_t time;
```

```
printf( "%4d/%02d/%02d %02d:%02d:%02d \n",  
        t.tm_year+1900, t.tm_mon+1, t.tm_mday,  
        t.tm_hour, t.tm_min, t.tm_sec );
```

```
2006/05/15 09:05:07
```

실습 5 정답

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <time.h>
5
6  void main( void )
7  {
8      char strTime[] = "2006/05/15 09:05:07";
9
10     char* token;
11     struct tm t;
12     time_t time;
13
14     token = strtok( strTime, "/" : " );
15     t.tm_year = atoi( token ) - 1900;
16
17     token = strtok( NULL, "/" : " );
18     t.tm_mon = atoi( token ) - 1;
19
20     token = strtok( NULL, "/" : " );
21     t.tm_mday = atoi( token );
22
23     token = strtok( NULL, "/" : " );
24     t.tm_hour = atoi( token );
25
26     token = strtok( NULL, "/" : " );
27     t.tm_min = atoi( token );
28
29     token = strtok( NULL, "/" : " );
30     t.tm_sec = atoi( token );
31
32     time = mktime( &t );
33
34     printf( "%4d/%02d/%02d %02d:%02d:%02d \n",
35            t.tm_year+1900, t.tm_mon+1, t.tm_mday,
36            t.tm_hour, t.tm_min, t.tm_sec );
37 }
```

실습 6

문제 6)

문자열에 저장된 시간을 읽어오기(atoi(), mktime() 함수)
정수값으로 변환하여 출력하시오

```
char strTime[] = "20060515090507"  
  
struct tm t;  
time_t time;
```

```
printf( "%4d/%02d/%02d %02d:%02d:%02d \n",  
        t.tm_year+1900, t.tm_mon+1, t.tm_mday,  
        t.tm_hour, t.tm_min, t.tm_sec );
```

```
2006/05/15 09:05:07
```


실습 6 정답

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include <time.h>
5
6 void main( void )
7 {
8     char strTime[] = "20060515090507";
9
10    struct tm t;
11    time_t time;
12
13    t.tm_sec = atoi( &strTime[12] );
14    strTime[12] = '\0';
15
16    t.tm_min = atoi( &strTime[10] );
17    strTime[10] = '\0';
18
19    t.tm_hour = atoi( &strTime[ 8] );
20    strTime[ 8] = '\0';
21
22    t.tm_mday = atoi( &strTime[ 6] );
23    strTime[ 6] = '\0';
24
25    t.tm_mon = atoi( &strTime[ 4] ) - 1;
26    strTime[ 4] = '\0';
27
28    t.tm_year = atoi( &strTime[0] ) - 1900;
29
30    time = mktime( &t );
31
32    printf( "%4d/%02d/%02d %02d:%02d:%02d \n",
33           t.tm_year+1900, t.tm_mon+1, t.tm_mday,
34           t.tm_hour, t.tm_min, t.tm_sec );
35 }
```

실습 7

문제 7)

문자열로 저장된 시간의 차이 구하기

mktime(), gmtime() 함수 사용

```
char strTime1[] = "9시 30분 21초";  
char strTime2[] = "9시 35분 30초";
```

```
struct tm t;  
time_t time1, time2;
```

```
printf( "%02d:%02d:%02d \n",  
        t.tm_hour, t.tm_min, t.tm_sec );
```

```
00:05:09
```

실습 7 정답

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <time.h>
5
6  void main( void )
7  {
8      char strTime1[] = "9시 30분 21초";
9      char strTime2[] = "9시 35분 30초";
10
11     struct tm t;
12     time_t time1, time2;
13
14     tzset();
15
16     t.tm_year = 70;
17     t.tm_mon = 0;
18     t.tm_mday = 1;
19     t.tm_hour = atoi( &strTime1[0] );
20     t.tm_min = atoi( &strTime1[4] );
21     t.tm_sec = atoi( &strTime1[11] );
22
23     time1 = mktime( &t );
24
25     t.tm_hour = atoi( &strTime2[0] );
26     t.tm_min = atoi( &strTime2[4] );
27     t.tm_sec = atoi( &strTime2[11] );
28
29     time2 = mktime( &t );
30
31     time2 -= time1;
32
33     t = *gmtime( &time2 );
34
35     printf( "%02d:%02d:%02d \n",
36            t.tm_hour, t.tm_min, t.tm_sec );
37 }
```

실습 8 문제 (1/3)

문제 8)

주차장 관리 프로그램 만들기

1. 입차

차량 번호, 차주 이름, 차주 핸드폰 번호를 입력 받고

(이미 있는 차량 번호 입력 시 다른 차량 번호 입력하게 함, 입차시 주차시간 등록)

2. 출력

-1 모든 출력 : 지금까지 입력된 모든 차량 정보 (차량번호, 차주 이름, 차주 핸드폰 번호), 주차 시간 및 주차 비용 출력

-2 개별 출력 : 차량번호를 입력 받아 입력 받은 차량번호의 정보, 주차 시간 및 주차 비용 출력

-0 메인 메뉴 : 메인 메뉴로 이동

3. 수정

차량번호를 입력 받아 입력 받은 차량번호의 정보 수정

* 수정 및 출력에서 없는 차량번호 입력 시 다시 입력 받게 함

* 최대 차량 번호 입력 개수 50개로 설정

* 주차 비용은 30분 이하 1000원, 30분 초과 1시간 이하 2000원,
1시간 초과 2시간 이하 3000원, 2시간 초과 5000원으로 설정

실습 8 문제 (2/3)

문제 8)

주차장 관리 프로그램 만들기

```
void mainMenu()
{
    puts("┌──────────────────┐");
    puts("    ||| car manual program |||      |");
    puts("    1.entrance  2.print  3.modify      |");
    puts("    4.exit      |");
    puts("    0.end      |");
    puts("└──────────────────┘");
    printf(" Select > ");
}

void outputMenu()
{
    puts("┌──────────────────┐");
    puts("    |||   selct print   |||      |");
    puts("    1. all print      |");
    puts("    2. each print     |");
    puts("    0. main menu      |");
    puts("└──────────────────┘");
    printf(" Select > ");
}
```

실습 8 문제 (3/3)

```

car manual program

1.entrance  2.print  3.modify

4.exit

0.end

Select > 1
input
car number : 1234
name : park
call number : 010-1111-1234

car manual program

1.entrance  2.print  3.modify

4.exit

0.end

Select > 2
select print

1. all print
2. each print
0. main menu

Select > 2
each print (car number) > 1111
car number | owner name | call number | parking time | parking cost
1234 | park | 010-1111-1234 | Thu Sep 1 08:49:59 2022 | 1000 |

car manual program

1.entrance  2.print  3.modify

4.exit

0.end

Select > 2
select print

1. all print
2. each print
0. main menu

Select > 2
each print (car number) > 1234
car number | owner name | call number | parking time | parking cost
1234 | park | 010-1111-1234 | Thu Sep 1 08:49:59 2022 | 1000 |

car manual program

1.entrance  2.print  3.modify

4.exit

0.end

Select > 2
select print

1. all print
2. each print
0. main menu

Select > 2
each print (car number) > 1234
car number | owner name | call number | parking time | parking cost
1234 | park | 010-1111-1234 | Thu Sep 1 08:49:59 2022 | 1000 |

car manual program

1.entrance  2.print  3.modify

4.exit

0.end

Select > 2
select print

1. all print
2. each print
0. main menu

Select > 2
each print (car number) > 1234
car number | owner name | call number | parking time | parking cost
1234 | park | 010-1111-1234 | Thu Sep 1 08:49:59 2022 | 1000 |
```

실습 8 정답 (1/8)

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <time.h>
4
5 #define MAX_CAR 50
6
7 struct Node
8 {
9     char car_num[20];
10    char name[20];
11    char phone[20];
12    time_t parking;
13 };
14
15 void mainMenu();
16 void outputMenu();
17
18 void inputFunc(struct Node* car);
19 void outputFunc(struct Node* car);
20 void outputFull(struct Node* car);
21 void outputSingle(struct Node* car);
22 void rmNewLine(char* tmpParkingTimeStr1, char* tmpParkingTimeStr2);
23 int parkingCost(double elapsedTime);
24
25 void modifyFunc(struct Node* car);
26 void outCarFunc(struct Node* car);
27
28 int main()
29 {
30     struct Node car[MAX_CAR];
31     int select,i;
32
33     for(i=0; i<MAX_CAR; i++)
34     {
35         strcpy(car[i].car_num, "");
36     }
```

실습 8 정답 (2/8)

```
38 while(1)
39 {
40     mainMenu();
41     fflush(stdin);
42     scanf("%d",&select);
43
44     switch(select)
45     {
46     case 1:
47         inputFunc(car);
48         break;
49     case 2:
50         outputFunc(car);
51         break;
52     case 3:
53         modifyFunc(car);
54         break;
55     case 4:
56         outCarFunc(car);
57         break;
58     case 0:
59         printf("prgram end.\n");
60         return 0;
61     default:
62         break;
63     }
64 }
65 return 0;
66 }
```


실습 8 정답 (3/8)

```
68 void mainMenu()  
69 {  
70     puts("┌──────────────────┐");  
71     puts("    ||| car manual program |||");  
72     puts("└──────────────────┘");  
73     puts("    1.entrance    2.print    3.modify");  
74     puts("└──────────────────┘");  
75     puts("    4.exit");  
76     puts("└──────────────────┘");  
77     puts("    0.end");  
78     puts("└──────────────────┘");  
79     printf(" Select > ");  
80 }  
81  
82 void outputMenu()  
83 {  
84     puts("┌──────────────────┐");  
85     puts("    |||   selct print   |||");  
86     puts("└──────────────────┘");  
87     puts("    1. all print");  
88     puts("└──────────────────┘");  
89     puts("    2. each print");  
90     puts("└──────────────────┘");  
91     puts("    0. main menu");  
92     puts("└──────────────────┘");  
93     printf(" Select > ");  
94 }
```

실습 8 정답 (4/8)

```
96 void inputFunc(struct Node* car)
97 {
98     time_t parkingTime;
99     int i, j;
100     int cnt=0;
101     puts("=== input ===");
102     for(i=0; i<MAX_CAR; i++)
103     {
104         if(strcmp(car[i].car_num, "") == 0) break;
105     }
106     printf("car number : "); scanf("%s", car[i].car_num);
107     for(j=0; j<MAX_CAR; j++) if(strcmp(car[i].car_num, car[j].car_num) == 0) cnt++;
108     if(cnt > 1)
109     {
110         strcpy(car[i].car_num, "");
111         puts("exist car number input other number again");
112         return;
113     }
114     printf("name : "); scanf("%s", car[i].name);
115     printf("call number : "); scanf("%s", car[i].phone);
116     car[i].parking = time(&parkingTime);
117 }
118
119
120 void outputFunc(struct Node* car)
121 {
122     int select;
123
124     while(1)
125     {
126         outputMenu();
127         fflush(stdin);
128         scanf("%d", &select);
129         switch(select)
130         {
131             case 1:
132                 outputFull(car);
133                 break;
134             case 2:
135                 outputSingle(car);
136                 break;
137             case 0:
138                 puts("back to main menu");
139                 return;
140             default:
141                 break;
142         }
143     }
144 }
```

실습 8 정답 (5/8)

```
146 void outputFull(struct Node* car)
147 {
148     int i;
149     char tmpParkingTimeStr1[26] = "";
150     char tmpParkingTimeStr2[26] = "";
151     time_t currentTime;
152     double elapsedTime;
153
154     printf("%10s | %8s | %13s | %26s | %8s | \n", "car number", "owner name", "call number", "parking time", "parking cost");
155     for(i=0; i<MAX_CAR; i++)
156     {
157         if(strcmp(car[i].car_num, "") != 0)
158         {
159             strcpy(tmpParkingTimeStr1, ctime(&car[i].parking));
160             rmNewLine(tmpParkingTimeStr1, tmpParkingTimeStr2); // ctime()에 의한 '\n' 제거
161
162             time(&currentTime);
163             elapsedTime = difftime(currentTime, car[i].parking);
164
165             printf("%10s | %8s | %13s | %26s | %8d | \n",
166                 car[i].car_num, car[i].name, car[i].phone, tmpParkingTimeStr2, parkingCost(elapsedTime));
167         }
168     }
169 }
170 }
171
172 void outputSingle(struct Node* car)
173 {
174     int i, cnt=0;
175     char tmpCarNum[20];
176     char tmpParkingTimeStr1[26] = "";
177     char tmpParkingTimeStr2[26] = "";
178     time_t currentTime;
179     double elapsedTime;
```

실습 8 정답 (6/8)

```
181 printf("each print (car number) > "); scanf("%s", tmpCarNum);
182 printf("%10s | %8s | %13s | %26s | %8s | \n", "car number", "owner name", "call number", "parking time", "parking cost");
183 for(i=0; i<MAX_CAR; i++)
184 {
185     if(strstr(car[i].car_num, tmpCarNum) != NULL)
186     {
187         strcpy(tmpParkingTimeStr1, ctime(&car[i].parking));
188         rmNewLine(tmpParkingTimeStr1, tmpParkingTimeStr2); // ctime()에 의한 '\n' 제거
189
190         time(&currentTime);
191         elapsedTime = difftime(currentTime, car[i].parking);
192
193         printf("%10s | %8s | %13s | %26s | %8d | \n",
194             car[i].car_num, car[i].name, car[i].phone, tmpParkingTimeStr2, parkingCost(elapsedTime));
195         cnt++;
196     }
197 }
198 if(cnt == 0) puts("couldn't find that car number");
199 }
200
201 void rmNewLine(char* tmpParkingTimeStr1, char* tmpParkingTimeStr2)
202 {
203     int i;
204     for(i=0; *(tmpParkingTimeStr1+i) != '\n'; i++)
205     {
206         *(tmpParkingTimeStr2+i) = *(tmpParkingTimeStr1+i);
207     }
208 }
209
210 int parkingCost(double elapsedTime)
211 {
212     if (elapsedTime <= 1800) return 1000; // 30분 이하
213     else if (elapsedTime > 1800 && elapsedTime <= 3600) return 2000; // 30분 초과 1시간 이하
214     else if (elapsedTime > 3600 && elapsedTime <= 7200) return 3000; // 1시간 초과 2시간 이하
215     else return 5000; // 2시간 초과
216 }
```

실습 8 정답 (7/8)

```
218 void modifyFunc(struct Node* car)
219 {
220     int i;
221     char tmpCarNum[20];
222
223     printf("modify (car number) > "); scanf("%s", tmpCarNum);
224     for(i=0; i<MAX_CAR; i++)
225     {
226         if(strcmp(tmpCarNum, car[i].car_num) == 0)
227         {
228             printf("owncer name : "); scanf("%s", car[i].name);
229             printf("call number : "); scanf("%s", car[i].phone);
230             printf(" %sth car number's information is changed\n", tmpCarNum);
231             return;
232         }
233     }
234     puts("couldn't find that car number");
235 }
236
237 void outCarFunc(struct Node* car)
238 {
239     int i;
240     char tmpCarNum[20];
241
242     printf("exist (car number) > "); scanf("%s", tmpCarNum);
243     for(i=0; i<MAX_CAR; i++)
244     {
245         if(strcmp(tmpCarNum, car[i].car_num) == 0)
246         {
247             strcpy(car[i].car_num, "");
248             printf(" %sth car number is exist\n", tmpCarNum);
249             return;
250         }
251     }
252     puts("couldn't find that car number");
253 }
```

실습 8 정답 (8/8)

```

car manual program
1.entrance  2.print  3.modify
4.exit
0.end

Select > 1
input
car number : 1234
name : park
call number : 010-1111-1234

car manual program
1.entrance  2.print  3.modify
4.exit
0.end

Select > 2
select print
1. all print
2. each print
0. main menu

Select > 1
car number | owner name | call number | parking time | parking cost
1234 | park | 010-1111-1234 | Thu Sep 1 08:49:59 2022 | 1000 |

select print
1. all print
2. each print
0. main menu

Select > 2
each print (car number) > 1111
car number | owner name | call number | parking time | parking cost
couldn't find that car number

select print
1. all print
2. each print
0. main menu

Select > 2
each print (car number) > 1234
car number | owner name | call number | parking time | parking cost
1234 | park | 010-1111-1234 | Thu Sep 1 08:49:59 2022 | 1000 |

select print
1. all print
```

질문 및 정리

