

학습활동 : 학생들의 성적을 입력받아 여러 기능을 추가한 프로그램

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

#define MAX_STUDENT 32 // 최대 학생수
#define MAX_LEN 10 // 최대 이름 길이

typedef struct STUDENT
{
    int number;
    char name[MAX_LEN];
    double grade;
} STUDENT;

// 학생 배열을 파라미터로 받아 전부 출력
void print_student(STUDENT student_arr[], int cnt)
{
    printf("\n");
    for (int i = 0; i < cnt; i++)
    {
        printf("학번: %d\n", student_arr[i].number);
        printf("이름: %s", student_arr[i].name);
        printf("학점: %.2lf\n\n", student_arr[i].grade);
    }
}

// 학생을 학생 배열에 추가하는 함수
// cnt의 주소를 받아옴(call by reference)
int add_student(STUDENT student_arr[], int *cnt)
{
    printf("학번을 입력하시오: ");
    scanf("%d", &student_arr[*cnt].number);
    if (student_arr[*cnt].number == -1) // 학번 입력 시 -1을 입력하면 함수도 -1을 리턴하고 종료
        return -1;
    rewind(stdin); // scanf 버퍼 비우기

    printf("이름을 입력하시오: ");
    fgets(student_arr[*cnt].name, MAX_LEN, stdin);

    printf("학점을 입력하시오(실수): ");
    scanf("%lf", &student_arr[(*cnt)++].grade); // 카운트가 가리키는 값을 인덱스로 사용 후 1만큼 증가

    rewind(stdin); // scanf 버퍼 비우기
    return 0;
}

// 버블 정렬 알고리즘을 이용해 학생 배열을 성적을 기준으로 오름차순 정렬
void sort_by_grade(STUDENT student_arr[], int cnt)
{

```

```
for (int i = 0; i < cnt - 1; i++)
{
    for (int j = 0; j < cnt - i - 1; j++)
    {
        if (student_arr[j].grade > student_arr[j + 1].grade)
        {
            // 성적을 기준으로 오름차순으로 교환
            STUDENT temp = student_arr[j];
            student_arr[j] = student_arr[j + 1];
            student_arr[j + 1] = temp;
        }
    }
}

// 버블 정렬 알고리즘을 이용해 학생 배열을 성적을 기준으로 내림차순 정렬
void sort_by_grade_desc(STUDENT student_arr[], int cnt)
{
    for (int i = 0; i < cnt - 1; i++)
    {
        for (int j = 0; j < cnt - i - 1; j++)
        {
            if (student_arr[j].grade < student_arr[j + 1].grade)
            {
                // 성적을 기준으로 내림차순으로 교환
                STUDENT temp = student_arr[j];
                student_arr[j] = student_arr[j + 1];
                student_arr[j + 1] = temp;
            }
        }
    }
}

// 학생들의 성적 중 최고점 혹은 최저점을 반환하는 함수
// mode 에 0 을 줄 경우 최고점, 1 을 줄 경우 최저점을 반환
double max_min_grade(int mode, STUDENT student_arr[], int cnt)
{
    double tmp = student_arr[0].grade;
    if (!mode)
    {
        for (int i = 1; i < cnt; i++)
        {
            if (tmp < student_arr[i].grade)
                tmp = student_arr[i].grade;
        }
        return tmp;
    }
    else
    {
        for (int i = 1; i < cnt; i++)
```

```
{
    if (tmp > student_arr[i].grade)
        tmp = student_arr[i].grade;
}
return tmp;
}
}

// 학생들의 성적 총합을 계산
double sum_grade(STUDENT student_arr[], int cnt)
{
    double sum = 0;

    for (int i = 0; i < cnt; i++)
    {
        sum += student_arr[i].grade;
    }

    return sum;
}

int main(void)
{
    STUDENT student_arr[MAX_STUDENT];
    int cnt = 0; // 배열의 숫자를 카운트
    char select = '0';

    while (1)
    {
        printf("\n===== \n");
        printf("\t[1]학생추가\t[2]오름차순정렬\t[3]내림차순정렬\n\t[4]최고점\t[5]최하점\t[6]성적평균\t[7]총인원출력\n");
        printf("===== \n");

        select = getche(); // 사용자로부터 메뉴버튼을 입력받음
        fputs("번 선택\n\0", stdout);

        if (select == '1')
        { // 1 번 : 학생 추가 모드
            if (add_student(student_arr, &cnt) == -1)
                break;
        }
        else if (select == '2')
        { // 2 번 : 성적 오름차순으로 정렬하여 출력
            sort_by_grade(student_arr, cnt);
            printf("\n 성적이 오름차순으로 정렬되었습니다.\n");
            print_student(student_arr, cnt);
        }
        else if (select == '3')
        { // 3 번 : 성적 내림차순으로 정렬하여 출력
            sort_by_grade_desc(student_arr, cnt);
            printf("\n 성적이 내림차순으로 정렬되었습니다.\n");
        }
    }
}
```

```
    print_student(student_arr, cnt);
}
else if (select == '4')
{ // 4 번 : 학생들의 성적 중 최고점을 출력
    printf("최고점은 %.2lf 입니다.\n", max_min_grade(0, student_arr, cnt));
}
else if (select == '5')
{ // 5 번 : 학생들의 성적 중 최하점을 출력
    printf("최하점은 %.2lf 입니다.\n", max_min_grade(1, student_arr, cnt));
}
else if (select == '6')
{ // 6 번 : 성적의 총합 및 평균을 출력
    printf("성적의 총합은 %.2lf, 평균은 %.2lf 입니다.\n", sum_grade(student_arr, cnt), sum_grade(student_arr, cnt) /
(double)cnt);
}
else if (select == '7')
{ // 7 번 : 현재 총 인원을 출력
    printf("현재 총 인원은 %d명 입니다.\n", cnt);
    print_student(student_arr, cnt);
}
else // 해당되지 않을 경우 프로그램 종료
    break;
}

return 0;
}
```

실행 결과

```

C:\Users\Wuser\Documents\W x + v
=====
[1] 학생 추가      [2] 오름차순 정렬  [3] 내림차순 정렬
[4] 최고점        [5] 최저점        [6] 성적 평균      [7] 총 인원 출력
=====
1 번 선택
학 번을 입력하시오: 20190895
이름을 입력하시오: 김찬영
학점을 입력하시오(실수): 4.3
=====
[1] 학생 추가      [2] 오름차순 정렬  [3] 내림차순 정렬
[4] 최고점        [5] 최저점        [6] 성적 평균      [7] 총 인원 출력
=====
1 번 선택
학 번을 입력하시오: 20190000
이름을 입력하시오: 허00
학점을 입력하시오(실수): 4.5
=====
[1] 학생 추가      [2] 오름차순 정렬  [3] 내림차순 정렬
[4] 최고점        [5] 최저점        [6] 성적 평균      [7] 총 인원 출력
=====
1 번 선택
학 번을 입력하시오: 20230000
이름을 입력하시오: 이00
학점을 입력하시오(실수): 3.5

```

1) 맨 처음 프로그램을 실행하고 1 번 메뉴를 선택하여 학생들을 3 명 추가하였다.

```

=====
[1] 학생 추가      [2] 오름차순 정렬  [3] 내림차순 정렬
[4] 최고점        [5] 최저점        [6] 성적 평균      [7] 총 인원 출력
=====
7 번 선택
현재 총 인원은 3명 입니다.

학 번: 20190895
이름: 김찬영
학점: 4.30

학 번: 20190000
이름: 허00
학점: 4.50

학 번: 20230000
이름: 이00
학점: 3.50

```

2) 정상적으로 학생들이 추가되었는지 7 번 메뉴를 선택하여 확인하였다. 학생들이 추가된 순서대로 정렬되어 출력된다.

```

=====
          [1]학생추가      [2]오름차순정렬 [3]내림차순정렬
          [4]최고점      [5]최하점      [6]성적평균      [7]총인원출력
=====
2번 선택

성적이 오름차순으로 정렬되었습니다.

학번: 20230000
이름: 이00
학점: 3.50

학번: 20190895
이름: 김찬영
학점: 4.30

학번: 20190000
이름: 허00
학점: 4.50

```

- 3) 2번 메뉴를 선택하여 학생들의 성적을 오름차순으로 출력한 결과이다. 정렬 시 버블 정렬 알고리즘을 사용했다. 출력만 오름차순으로 정렬하는 것이 아닌, 학생들의 데이터가 저장된 구조체 배열 `student_arr[]` 내부의 순서 자체를 바꾸도록 설정하였다.

```

=====
          [1]학생추가      [2]오름차순정렬 [3]내림차순정렬
          [4]최고점      [5]최하점      [6]성적평균      [7]총인원출력
=====
3번 선택

성적이 내림차순으로 정렬되었습니다.

학번: 20190000
이름: 허00
학점: 4.50

학번: 20190895
이름: 김찬영
학점: 4.30

학번: 20230000
이름: 이00
학점: 3.50

```

- 4) 3번 메뉴를 선택하여 학생들의 성적을 내림차순으로 출력한 결과이다, 2번 메뉴와 마찬가지로 버블 정렬 알고리즘을 사용하여 실제 배열의 순서를 재배치하는 방식으로 진행하였다.

```
=====
[1]학생추가    [2]오름차순정렬 [3]내림차순정렬
[4]최고점      [5]최하점      [6]성적평균    [7]총인원출력
=====
```

```
4번 선택
최고점은 4.50 입니다.
```

```
=====
[1]학생추가    [2]오름차순정렬 [3]내림차순정렬
[4]최고점      [5]최하점      [6]성적평균    [7]총인원출력
=====
```

```
5번 선택
최하점은 3.50 입니다.
```

- 5) 4 번 최고점 출력과 5 번 최하점 출력 모드를 실행한 결과이다. 사용된 알고리즘은 기본적으로 temp 를 사용하여 최댓값/최솟값을 찾는 데 사용되는 방식을 사용했고 정상적으로 최고점인 4.5, 최하점인 3.5 를 출력하는 모습을 볼 수 있다.

```
=====
[1]학생추가    [2]오름차순정렬 [3]내림차순정렬
[4]최고점      [5]최하점      [6]성적평균    [7]총인원출력
=====
```

```
6번 선택
성적의 총합은 12.30, 평균은 4.10 입니다.
```

- 6) 6 번 메뉴를 선택하면 성적의 총합과 평균을 출력하도록 설정하였다.

```
=====
[1]학생추가    [2]오름차순정렬 [3]내림차순정렬
[4]최고점      [5]최하점      [6]성적평균    [7]총인원출력
=====
```

```
1번 선택
학번을 입력하시오: -1
```

```
PS C:\Users\User\Documents\vscode> █
```

- 7) 학번 입력 시 -1 을 입력하면 프로그램이 종료되는 모습을 확인할 수 있다.

자신이 코딩하고 싶은 것 : 구조체 활용

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

// 날짜를 나타내는 구조체
struct Date
{
    int year;
    int month;
    int day;
};

// 상품을 나타내는 구조체
struct Product
{
    struct Date registrationDate; // 등록된 날짜
    char name[50];                // 상품의 이름 (최대 50 자)
    char type[20];                // 상품의 종류 (최대 20 자)
};

// 사용자로부터 상품 정보를 입력받는 함수
void inputProductInfo(struct Product *product)
{
    // 날짜 입력
    printf("등록된 날짜를 입력하세요 (년 월 일): ");
    scanf("%d %d %d", &product->registrationDate.year, &product->registrationDate.month, &product->registrationDate.day);

    // 버퍼 비우기
    while (getchar() != '\n')
        ;

    // 이름 입력
    printf("상품의 이름을 입력하세요: ");
    fgets(product->name, sizeof(product->name), stdin);
    product->name[strcspn(product->name, "\n")] = '\0'; // 개행 문자 제거

    // 종류 입력
    printf("상품의 종류를 입력하세요: ");
    fgets(product->type, sizeof(product->type), stdin);
    product->type[strcspn(product->type, "\n")] = '\0'; // 개행 문자 제거
}

int main()
{
    struct Product myProduct;

    // 사용자로부터 상품 정보 입력 받기
    inputProductInfo(&myProduct);

    // 정보 출력
```



```
    printf("등록된 날짜: %d 년 %d 월 %d 일\n", myProduct.registrationDate.year, myProduct.registrationDate.month,
myProduct.registrationDate.day);
    printf("상품 이름: %s\n", myProduct.name);
    printf("상품 종류: %s\n", myProduct.type);

    return 0;
}
```

실행 결과

```
PS C:\Users\User\Documents\vscode> cd "c:\Users\User\Documents\vscode\" ; if ($?) { gcc
a.c -o a } ; if ($?) { .\a }
등록된 날짜를 입력하세요 (년 월 일): 2023 11 04
상품의 이름을 입력하세요: 맥북에어
상품의 종류를 입력하세요: 노트북
등록된 날짜: 2023년 11월 4일
상품 이름: 맥북에어
상품 종류: 노트북
PS C:\Users\User\Documents\vscode> █
```

P.585 : 01 번 문제 소스코드

```
#include <stdio.h>

typedef enum GENRE // 열거형을 typedef 로 선언
{
    COMIC,
    SF,
    DOCU,
    DRAMA
} GENRE;

typedef struct BOOK // 구조체를 typedef 로 선언
{
    char title[100];
    char author[100];
    GENRE type; // 열거형 데이터 GENRE 가 구조체 내에 포함됨
} BOOK;

int main()
{
    char *genre_name[] = {"COMIC", "SF", "DOCU", "DRAMA"};

    BOOK a = {"바람과 함께 사라지다", "마가렛 미첼", SF}; // 구조체 변수 선언

    printf("{%s, %s, %s}", a.title, a.author, genre_name[a.type]); // 구조체 멤버를 출력

    return 0;
}
```

P.585 : 01 번 문제 실행 결과

```
PS C:\Users\user\Documents\vscode> cd "c:\Users\user\Documents\vscode\" ; if ($?) { gcc 585_1.c -o 585_1 } ; if ($?) { .\585_1
}
{바람과 함께 사라지다, 마가렛 미첼, SF}
PS C:\Users\user\Documents\vscode> □
```

P.585 : 02 번 문제 소스코드

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

// 열거형 정의
typedef enum GENRE
{
    COMIC,
    SF,
    DOCU,
    DRAMA
} GENRE;

// 구조체 정의
typedef struct BOOK
{
    char title[100];
    char author[100];
    GENRE type;
} BOOK;

// 작가가 동일한지 확인하는 함수
int equal_author(BOOK b1, BOOK b2)
{
    // strcmp 함수를 사용하여 작가 이름을 비교
    if (!strcmp(b1.author, b2.author))
    {
        return 1; // 같으면 1 반환
    }
    else
    {
        return 0; // 다르면 0 반환
    }
}

int main()
{
    // 장르 이름 배열
    char *genre_name[] = {"COMIC", "SF", "DOCU", "DRAMA"};

    // 책 두 권의 정보를 가진 구조체 변수 선언 및 초기화
    BOOK b1 = {"노인과 바다", "헤밍웨이", DRAMA};
    BOOK b2 = {"누구를 위하여 종을 울리나", "헤밍웨이", DRAMA};

    // 책 정보 출력
```

```

printf("{%s, %s, %s}\n", b1.title, b1.author, genre_name[b1.type]);
printf("{%s, %s, %s}\n", b2.title, b2.author, genre_name[b2.type]);

// 작가가 동일한지 확인하고 결과 출력
printf("equal_author()의 반환값: %d", equal_author(b1, b2));

return 0;
}

```

P.585 : 02 번 문제 실행 결과

두 책의 저자를 다르게 설정하여 코드를 실행하면 equal_author()의 반환값이 0 이 된다. 하지만 두번째 사진과 같이 두 책의 저자를 같게 초기화하고 실행하면 반환값이 1 인 것을 확인할 수 있다.

```

PS C:\Users\User\Documents\vscode> cd "c:\Users\User\Documents\vscode" ; if ($?) { gcc 585_2.c -o 585_2 } ; if ($?) { .\585_2
}
{노인과 바다, 헤밍웨이, DRAMA}
{누구를 위하여 종을 울리나, 헤밍웨이!, DRAMA}
equal_author()의 반환값: 0
PS C:\Users\User\Documents\vscode> 

```

(* b1.author : 헤밍웨이, b2.author : 헤밍웨이!로 다르게 설정)

```

PS C:\Users\User\Documents\vscode> cd "c:\Users\User\Documents\vscode" ; if ($?) { gcc 585_2.c -o 585_2 } ; if ($?) { .\585_2
}
{노인과 바다, 헤밍웨이, DRAMA}
{누구를 위하여 종을 울리나, 헤밍웨이, DRAMA}
equal_author()의 반환값: 1
PS C:\Users\User\Documents\vscode> 

```

(* b1.author : 헤밍웨이, b2.author : 헤밍웨이로 동일하게 설정)

P.589 : 14 번 문제 소스코드

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

#define MAX_MP3_CNT 3 // MP3 파일 저장용량

typedef enum GENRE
{
    KPOP,
    POP,
    CLASSIC,
    MOVIEMUSIC
} GENRE;

char *genre_name[] = {"KPOP", "POP", "CLASSIC", "MOVIEMUSIC"};

typedef struct MP3
{
    char name[32];
    char artist[32];
    char addr[64];
    GENRE genre;
} MP3;

int main()
{
    MP3 music[MAX_MP3_CNT];
    int select = 0;
    char buf[32];
    int p = 0;

    while (1)
    {
        // 메뉴 출력
        printf("\n===== \n");
        printf("1.추가  2.출력  3.검색  4.종료\n");
        printf("===== \n");

        // 사용자에게 정수 입력 받기
        printf("정수값을 입력하시오>>");
        scanf("%d", &select);

        // 입력 버퍼 초기화
        rewind(stdin);
```

```
// 기능 선택에 따른 분기
if (select == 1)
{
    // 추가 기능 선택
    if (p >= MAX_MP3_CNT)
    {
        // 저장용량이 가득 찬 경우
        printf("저장용량이 가득 찼습니다\n");
        continue;
    }
    else
    {
        // 음악 정보 입력
        printf("제목>>");
        gets(music[p].name);
        printf("가수>>");
        gets(music[p].artist);
        printf("위치>>");
        gets(music[p].addr);
        printf("장르(0:가요, 1:팝, 2:클래식, 3:영화음악)>>");
        scanf("%d", &music[p].genre);
        p++; // 현재 저장된 음악 개수
    }
}
else if (select == 2)
{
    // 출력 기능 선택
    // 저장된 음악 정보 출력
    for (int i = 0; i < p; i++)
    {
        printf("\nNO.%02d\n 제목:%s\n 가수:%s\n 위치:%s\n 장르:%s\n",
            i, music[i].name, music[i].artist, music[i].addr, genre_name[music[i].genre]);
    }
}
else if (select == 3)
{
    // 검색 기능 선택
    // 제목으로 음악 검색
    printf("제목 검색>>");
    gets(buf);
    for (int i = 0; i < p; i++)
    {
        if (!strcmp(buf, music[i].name))
        {
            printf("\nNO.%02d\n 제목:%s\n 가수:%s\n 위치:%s\n 장르:%s\n",
```

```
        i, music[i].name, music[i].artist, music[i].addr, genre_name[music[i].genre]);
    }
}
else if (select == 4)
{
    // 종료 기능 선택
    break;
}
// 사용자 선택 항목 초기화
select = 0;
}

return 0;
}
```

P.589 : 14 번 문제 실행 결과

```

=====
1.추가 2.출력 3.검색 4.종료
=====
정수값을 입력하시오>>1
제목>>야생화
가수>>박효신
위치>>C:\Music
장르(0:가요, 1:팝, 2:클래식, 3:영화음악)>>0

=====
1.추가 2.출력 3.검색 4.종료
=====
정수값을 입력하시오>>1
제목>>See You Again
가수>>찰리 푸스
위치>>C:\Music
장르(0:가요, 1:팝, 2:클래식, 3:영화음악)>>1

=====
1.추가 2.출력 3.검색 4.종료
=====
정수값을 입력하시오>>1
제목>>사계
가수>>비발디
위치>>C:\Music\Classic\
장르(0:가요, 1:팝, 2:클래식, 3:영화음악)>>2

=====
1.추가 2.출력 3.검색 4.종료
=====
정수값을 입력하시오>>1
저장용량이 가득 찼습니다

=====
1.추가 2.출력 3.검색 4.종료
=====
정수값을 입력하시오>>2

NO.00
제목:야생화
가수:박효신
위치:C:\Music
장르:KPOP

NO.01
제목:See You Again
가수:찰리 푸스
위치:C:\Music
장르:POP

NO.02
제목:사계
가수:비발디
위치:C:\Music\Classic\
장르:CLASSIC

=====
1.추가 2.출력 3.검색 4.종료
=====
정수값을 입력하시오>>3
제목 검색>>야생화

NO.00
제목:야생화
가수:박효신
위치:C:\Music
장르:KPOP

=====
1.추가 2.출력 3.검색 4.종료
=====
정수값을 입력하시오>>3
제목 검색>>See You Again

NO.01
제목:See You Again
가수:찰리 푸스
위치:C:\Music
장르:POP

=====
1.추가 2.출력 3.검색 4.종료
=====
정수값을 입력하시오>>3
제목 검색>>사계

NO.02
제목:사계
가수:비발디
위치:C:\Music\Classic\
장르:CLASSIC

=====
1.추가 2.출력 3.검색 4.종료
=====
정수값을 입력하시오>>

```

처음에 3 개의 음악을 추가하였고, 4 번째 음악을 추가하려니 용량이 가득 찼다는 말과 함께 추가되지 않는다. 또한, 모든 음악에 대한 정보를 출력하는 기능도 정상 동작하였고 제목으로 검색하여 음악을 찾는 기능도 동작함을 확인하였다. 4 번 항목을 고를 경우 프로그램이 종료된다.

P.589 : 15 번 문제 소스코드

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

#define MAX_BOOK_CNT 3 // 최대 책 저장

typedef enum ISOUT
{
    EXIST,
    OUT
} ISOUT;

char *enum_isout[] = {"현재 보유중", "대출됨"};

typedef struct BOOK
{
    int num;
    char name[32];
    char author[32];
    ISOUT isout;
} BOOK;

// 도서 정보 출력하는 함수
void print_book(BOOK book)
{
    printf("\n 도서번호:%d\n 도서명:%s\n 저자:%s\n 대출여부:%s\n", book.num, book.name, book.author,
enum_isout[book.isout]);
}

int main()
{
    BOOK book[MAX_BOOK_CNT];
    int select = 0;
    char buf[32];
    int cnt = 0; // 총 저장된 책의 수
    int ex = 0; // 대출되지 않은 책의 수

    while (1)
    {
        // 메뉴 출력
        printf("\n=====");
        printf("1.도서 번호로 책 찾기\n2.저자 이름으로 책 찾기\n3.제목으로 책 찾기\n4.새로운 책 추가\n5.도서관이
소장한 도서의 수 표시\n");
        printf("=====");

        // 사용자에게 정수 입력 받기
        printf("메뉴 중 하나를 선택>>");
        scanf("%d", &select);
```

```
// 입력 버퍼 초기화
rewind(stdin);

if (select == 1)
{ // 1.도서 번호로 책 찾기
    printf("도서 번호 입력>>");
    gets(buf);
    for (int i = 0; i < cnt; i++)
    {
        if (atoi(buf) == book[i].num)
        {
            print_book(book[i]);
            if (book[i].isout)
            {
                printf("대출 불가!\n");
            }
            else
            {
                printf("도서를 대출합니다\n");
                book[i].isout = OUT;
            }
        }
    }
}
else if (select == 2)
{ // 2.저자명으로 책 찾기
    printf("저자 이름 입력>>");
    gets(buf);
    for (int i = 0; i < cnt; i++)
    {
        if (!strcmp(buf, book[i].author))
        {
            print_book(book[i]);
            if (book[i].isout)
            {
                printf("대출 불가!\n");
            }
            else
            {
                printf("도서를 대출합니다\n");
                book[i].isout = OUT;
            }
        }
    }
}
else if (select == 3)
{ // 3.도서명으로 책 찾기
    printf("도서 제목 입력>>");
    gets(buf);
    for (int i = 0; i < cnt; i++)
```

```
{
    if (!strcmp(buf, book[i].name))
    {
        print_book(book[i]);
        if (book[i].isout)
        {
            printf("대출 불가!\n");
        }
        else
        {
            printf("도서를 대출합니다\n");
            book[i].isout = OUT;
        }
    }
}

}

else if (select == 4)
{ // 4.새로운 도서 추가
    if (cnt >= MAX_BOOK_CNT)
    {
        // 도서 저장한도가 가득 찬 경우
        printf("도서 추가 불가!\n");
        continue;
    }
    else
    {
        // 도서 정보 입력
        printf("도서번호>>");
        scanf("%d", &book[cnt].num);
        rewind(stdin);
        printf("도서명>>");
        gets(book[cnt].name);
        printf("저자>>");
        gets(book[cnt].author);
        book[cnt].isout = EXIST;
        cnt++; // 현재 저장된 도서의 총 개수
    }
}

else if (select == 5)
{
    for (int i = 0; i < cnt; i++)
    {
        if (book[i].isout != OUT)
        { // 현재 도서관이 보유중일 경우
            ex++;
        }
        else
            continue;
    }
    printf("총 도서 수: %d 권\n 현재 존재하는 도서: %d 권\n", cnt, ex);
}
```

```

    ex = 0;
    select = 0;
}

return 0;
}

```

P.589 : 15 번 문제 실행 결과

```

=====
1.도서 번호로 책 찾기
2.저자 이름으로 책 찾기
3.제목으로 책 찾기
4.새로운 책 추가
5.도서관이 소장한 도서의 수 표시
=====

```

```

메뉴 중 하나를 선택>>4
도서번호>>1001
도서명>>나무야나무야
저자>>신영복

```

```

=====
1.도서 번호로 책 찾기
2.저자 이름으로 책 찾기
3.제목으로 책 찾기
4.새로운 책 추가
5.도서관이 소장한 도서의 수 표시
=====

```

```

메뉴 중 하나를 선택>>4
도서번호>>2001
도서명>>최애의아이
저자>>아카사카 아카

```

```

=====
1.도서 번호로 책 찾기
2.저자 이름으로 책 찾기
3.제목으로 책 찾기
4.새로운 책 추가
5.도서관이 소장한 도서의 수 표시
=====

```

```

메뉴 중 하나를 선택>>5
총 도서 수: 2권
현재 존재하는 도서: 2권

```

```

=====
1.도서 번호로 책 찾기
2.저자 이름으로 책 찾기
3.제목으로 책 찾기
4.새로운 책 추가
5.도서관이 소장한 도서의 수 표시
=====

```

```

메뉴 중 하나를 선택>>3
도서 제목 입력>>최애의아이

```

```

도서번호:2001
도서명:최애의아이
저자:아카사카 아카
대출여부:현재 보유중
도서를 대출합니다

```

```

=====
1.도서 번호로 책 찾기
2.저자 이름으로 책 찾기
3.제목으로 책 찾기
4.새로운 책 추가
5.도서관이 소장한 도서의 수 표시
=====

```

```

메뉴 중 하나를 선택>>2
저자 이름 입력>>신영복

```

```

도서번호:1001
도서명:나무야나무야
저자:신영복
대출여부:현재 보유중
도서를 대출합니다

```

```

=====
1.도서 번호로 책 찾기
2.저자 이름으로 책 찾기
3.제목으로 책 찾기
4.새로운 책 추가
5.도서관이 소장한 도서의 수 표시
=====

```

```

메뉴 중 하나를 선택>>5
총 도서 수: 2권
현재 존재하는 도서: 0권

```

```

=====
1.도서 번호로 책 찾기
2.저자 이름으로 책 찾기
3.제목으로 책 찾기
4.새로운 책 추가
5.도서관이 소장한 도서의 수 표시
=====

```

```

메뉴 중 하나를 선택>>1
도서 번호 입력>>2001

```

```

도서번호:2001
도서명:최애의아이
저자:아카사카 아카
대출여부:대출됨
대출 불가!

```

```

=====
1.도서 번호로 책 찾기
2.저자 이름으로 책 찾기
3.제목으로 책 찾기
4.새로운 책 추가
5.도서관이 소장한 도서의 수 표시
=====

```

```

메뉴 중 하나를 선택>>

```

왼쪽부터 차례대로 실행 결과를 나열하였다. 먼저 책 2 권을 추가한 뒤, 5 번 메뉴를 실행하여 정상적으로 책 2 권이 추가되었고, 도서관에 있는 것을 확인할 수 있었다. 두번째 사진에는 제목으로 책을 검색하고, 저자로 책을 검색하여 책을 대출하는 장면이다. 2 권을 대출한 뒤, 5 번 항목을 실행하면 2 권 모두 대출됨을 확인할 수 있다. 세번째 사진에는 도서 번호로 책을 검색하는 장면이다. 앞서 대출된 책을 대출하려 하자 대출 불가라는 안내문이 뜨는 것을 확인할 수 있다.

구조체 도전 문제 소스 코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <time.h> // time() 함수를 사용하기 위해
#define SIZE 100

// 문제 구조체 정의
struct QUESTION {
    char question[SIZE];
    char item1[SIZE];
    char item2[SIZE];
    char item3[SIZE];
    char item4[SIZE];
    int solution;
};

struct QUESTION bank[10] = {
    {"임베디드 장치에 가장 적합한 프로그래밍 언어는?",
     "1.Python",
     "2.JAVA",
     "3.C",
     "4.Javascript",
     3},
    {"서로 다른 자료형을 모을 수 있는 구조는?",
     "1.배열",
     "2.변수",
     "3.구조체",
     "4.포인터",
     3},
    {"포인터의 크기는?",
     "1.32 비트",
     "2.64 비트",
     "3.컴파일러에 따라 다름",
     "4.고정된 크기 없음",
     4},
    {"동적 메모리 할당을 위해 사용되는 함수는?",
     "1.malloc",
     "2.calloc",
     "3.realloc",
     "4.free",
     1},
    {"C 언어의 대표적인 컴파일러는?",
     "1.GCC",
     "2.Visual Studio",
```

```
"3.Clang",
"4.Intel Compiler",
1},
{"헤더 파일의 확장자는?",
"1.h",
"2.c",
"3.cpp",
"4.exe",
1},
{"C 언어에서 사용되는 논리 연산자 'AND'는?",
"1.&&",
"2.||",
"3.!",
"4.&",
1},
{"반복문 중 무한 루프를 만들 때 사용되는 예약어는?",
"1.while",
"2.do-while",
"3.for",
"4.infinite",
4},
{"C 언어에서 변수를 메모리에서 삭제하는 키워드는?",
"1.delete",
"2.dispose",
"3.free",
"4.remove",
3},
{"함수의 반환형이 void 인 경우, return 문을 사용할 수 있을까?",
"1.가능하다",
"2.불가능하다",
"3.컴파일 에러 발생",
"4.경고만 발생",
2}
};

int main(void) {
    int select, i, score = 0; // 점수를 표시하는 score 변수

    srand((unsigned int)time(NULL)); // 시드값 실행될 때마다 초기화

    int question_seq[10]; // 결과를 저장할 배열

    // 랜덤한 순서로 문제 번호 생성
    for (int i = 0; i < 10; i++) {
        question_seq[i] = rand() % 10;
```

```
// 중복 체크
for (int j = 0; j < i; j++) {
    if (question_seq[i] == question_seq[j]) {
        i--; // 중복된 번호면 다시 생성
    }
}

// 생성된 문제 번호 출력
printf("문제 번호 순서 : ");
for (i = 0; i < 10; i++) {
    printf("%d ", question_seq[i]);
}
printf("\n");

// 문제 출력 및 사용자 답변 처리
for (i = 0; i < 10; i++) {
    printf("%s\n", bank[question_seq[i]].question);
    printf("%s  ", bank[question_seq[i]].item1);
    printf("%s  ", bank[question_seq[i]].item2);
    printf("%s  ", bank[question_seq[i]].item3);
    printf("%s  ", bank[question_seq[i]].item4);
    scanf("%d", &select);

    // 정답 체크
    if (select == bank[question_seq[i]].solution)
    {
        printf("맞았습니다\n\n");
        score++;
    }
    else
        printf("틀렸습니다\n\n");

    printf("현재 점수는 [%2d] 입니다\n\n", score); // 현재 점수 출력
}

return 0;
}
```

구조체 도전 문제 실행 결과

Microsoft Visual Studio 디버그 콘솔

```
문제 번호 순서 : 6 7 9 0 8 4 5 1 3 2
C 언어에서 사용되는 논리 연산자 'AND'는?
1.&& 2.|| 3.! 4.& 1
맞았습니다
```

현재 점수는 [1] 입니다

```
반복문 중 무한 루프를 만들 때 사용되는 예약어는?
1.while 2.do-while 3.for 4.infinite 4
맞았습니다
```

현재 점수는 [2] 입니다

```
함수의 반환형이 void인 경우, return 문을 사용할 수 있을까?
1.가능하다 2.불가능하다 3.컴파일 에러 발생 4.경고만 발생 2
맞았습니다
```

현재 점수는 [3] 입니다

```
임베디드 장치에 가장 적합한 프로그래밍 언어는?
1.Python 2.JAVA 3.C 4.Javascript 3
맞았습니다
```

현재 점수는 [4] 입니다

```
C 언어에서 변수를 메모리에서 삭제하는 키워드는?
1.delete 2.dispose 3.free 4.remove 3
맞았습니다
```

현재 점수는 [5] 입니다

```
C 언어의 대표적인 컴파일러는?
1.GCC 2.Visual Studio 3.Clang 4.Intel Compiler 1
맞았습니다
```

현재 점수는 [6] 입니다

현재 점수는 [7] 입니다

```
서로 다른 자료형을 모을 수 있는 구조는?
1.배열 2.변수 3.구조체 4.포인터 3
맞았습니다
```

현재 점수는 [8] 입니다

```
동적 메모리 할당을 위해 사용되는 함수는?
1.malloc 2 calloc 3.realloc 4.free 1
맞았습니다
```

현재 점수는 [9] 입니다

```
포인터의 크기는?
1.32비트 2.64비트 3.컴파일러에 따라 다름 4.고정된 크기 없음 4
맞았습니다
```

현재 점수는 [10] 입니다

```
C:\Users\Administrator\source\repos\Project1\wx64\Debug\Project1.exe(프로젝트1)
이 창을 닫으려면 아무 키나 누르세요...
```

(1) 문제를 10 문제 정도 추가해보라

문제를 총 10 문제로 추가하여 소스코드에 작성하였음

(2) 위의 프로그램에서 전체 문제 중에서 몇 문제를 맞추었는지를 표시하는 기능을 추가하라.

문제를 내고 사용자에게 정답을 입력받으면 맞췄는지 틀렸는지 확인한 다음, score 변수를 출력하여 사용자의 현재 점수를 계속 출력하도록 하였다.

(3) 문제가 랜덤하게 제시되도록 프로그램을 수정해보자. 한번 제시한 문제가 다시 나오면 안 된다.

Srand(time(NULL))함수로 프로그램이 매번 실행될 때 마다 랜덤한 시드값을 가지도록 하였으며, Rand()함수와 이중 반복문을 활용하여 중복 없이 배열에 0 부터 9 까지 채워넣도록 프로그래밍 하였다.

Question_seq[]배열에 0~9 의 난수를 집어넣은 다음, 이전의 인덱스 값들과 모두 비교하여 중복이 있는 경우 i의 값을 감소시킨다. i의 값이 감소된 상태로 j loop가 끝나게 되면 다시 i의 값이 이전과 같은 단계에서 진행된다. 즉, 중복이 발생된 자리의 숫자를 다시 난수를 발생해 배열에 저장하는 결과를 얻을 수 있다. 이렇게 반복하다 보면, 중복이 없는 배열이 생성된다. 그 다음, 반복문을 다시 사용하여 이 question_seq[]배열의 숫자대로 문제를 출력하면 원하는 답을 얻을 수 있다.