

//C프로그래밍응용(001) REPORT1 20230822 김민철

//자신이 코딩하고 싶은 것: 구조체 활용(이름과 나이를 입력받아 배열에 저장하고 출력)

//프로그램 설계 사항

- 1) 'struct Person' 구조체를 정의하여 이름과 나이를 저장할 수 있는 데이터 구조를 생성합니다.
- 2) 'main' 함수가 실행도리 때, 초기화된 'people' 배열과 'count' 변수를 선언합니다. 'people' 배열은 최대 10명의 사람 정보를 저장할 수 있습니다.
- 3) 'for' 반복문을 사용하여 사용자로부터 이름과 나이를 입력받습니다. 입력 시, "이름"과 "나이"를 입력하고, 나이에 -1을 입력하면 입력을 종료합니다.
4. 입력된 사람 정보는 'people' 배열에 저장되며, 'count' 변수는 입력된 사람의 수를 추적합니다.
5. 입력이 완료되면, "입력된 정보"를 출력하고, 'for' 반복문을 사용하여 'people' 배열에 저장된 정보를 출력합니다.

//코드

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
struct Person
```

```
{
```

```
    char name[50];
```

```
    int age;
```

```
};
```

```
int main()
```

```
{
```

```
    printf("[컴퓨터공학과, 김민철, 20230822]\n");
```

```
    printf("\n");
```

```
    struct Person people[10];
```

```
    int count = 0;
```

```
    for (int i = 0; i < 10; i++)
```

```
    {
```

```
        printf("-----\n입력을 그만두려면 나이에 -1을 입력하시  
오.\n");
```

```
        printf("이름: ");
```

```
        scanf("%s", people[i].name);
```

```
        printf("나이: ");
```

```
        scanf("%d", &people[i].age);
```

```
        printf("-----\n");
```

```
        printf("\n");
```

```

        if (people[i].age == -1)
        {
            break;
        }

        count++;
    }

    printf("\n입력된 정보:\n");

    for (int i = 0; i < count; i++)
    {
        printf("이름: %s, 나이: %d\n", people[i].name, people[i].age);
    }

    return 0;
}

```

//실행 결과

```

[컴퓨터공학과, 김민철, 20230822]

-----
입력을 그만두려면 나이에 -1을 입력하십시오.
이름: 김민철
나이: 20
-----

입력을 그만두려면 나이에 -1을 입력하십시오.
이름: 박미르
나이: 22
-----

입력을 그만두려면 나이에 -1을 입력하십시오.
이름: 김철수
나이: 34
-----

입력을 그만두려면 나이에 -1을 입력하십시오.
이름: 홍길동
나이: 55
-----

입력을 그만두려면 나이에 -1을 입력하십시오.
이름: .
나이: -1
-----

입력된 정보:
이름: 김민철, 나이: 20
이름: 박미르, 나이: 22
이름: 김철수, 나이: 34
이름: 홍길동, 나이: 55

C:\Users\user\source\repos\Project163\x64\Debug\Project163.exe(프로세스 17240개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

//구조체 도전 문제 - 4지 선택 퀴즈 프로그램

//프로그램 설계 사항

- 1) 'struct QUESTION' 구조체를 정의하여 퀴즈 질문, 선택지 및 정답을 저장할 수 있는 데이터 구조를 생성합니다.
- 2) 'main' 함수가 실행될 때, 초기화된 'bank' 배열과 다른 변수들을 선언합니다.
- 3) 'for' 반복문을 사용하여 12번의 퀴즈를 제시하고 사용자로부터 답을 입력받습니다. 무작위로 선택된 퀴즈는 중복되지 않도록 관리됩니다.
- 4) 답을 입력받을 때, 사용자가 1부터 4 사이의 값을 입력하도록 하기 위해 'while (1)' 루프와 'if' 문을 사용합니다. 올바르지 않은 값(1 미만 또는 4 초과)을 입력하면 사용자에게 올바른 범위의 값을 입력하도록 요청합니다.
- 5) 사용자의 답을 검사하여 정답 여부를 판별하고, 그에 따라 메시지를 표시합니다.
- 6) 프로그램이 종료되면, 사용자가 몇 개의 문제를 맞췄는지를 출력합니다.

//코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define SIZE 100
```

```
struct QUESTION {
    char question[SIZE];
    char item1[SIZE];
    char item2[SIZE];
    char item3[SIZE];
    char item4[SIZE];
    int solution;
};
```

```
struct QUESTION bank[100] = {
    {"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},
    {"TCP와 UDP 프로토콜 중 어떤 프로토콜이 연결 지향적이고 안정성을 제공하는가?", "1. TCP", "2. UDP", "3. HTTP", "4. IP", 1},
    {"SQL에서 'SELECT'는 무엇을 수행하는데 사용되는가?", "1. 데이터 삽입", "2. 데이터 삭제", "3. 데이터 조회", "4. 데이터 업데이트", 3},
    {"HTML, CSS 및 JavaScript 중 어떤 언어가 웹 페이지의 디자인을 제어하는 데 사용되는가?", "1. HTML", "2. CSS", "3. JavaScript", "4. 모두", 2},
    {"Big O 표기법에서 'O(n)'는 무엇을 나타내는가?", "1. 선형 시간 복잡도", "2. 로그 시간 복잡도", "3. 지수 시간 복잡도", "4. 상수 시간 복잡도", 1},
};
```

```

        {"운영 체제의 주요 역할 중 하나는 무엇인가?", "1. 웹 개발", "2. 파일 관리", "3. 음악 재생",
"4. 게임 실행", 2},
        {"C++에서 \'cout\'과 \'cin\'을 위해 사용되는가?", "1. 문자열 처리", "2. 파일 입출력", "3.
표준 입출력", "4. 그래픽 처리", 3},
        {"데이터베이스에서 \'SQL\'은 무엇을 나타내는가?", "1. 표준 쿼리 언어", "2. 분산 파일 시스
템", "3. 그래픽 디자인 언어", "4. 소프트웨어 개발 도구", 1},
        {"이진 검색(Binary Search) 알고리즘은 무엇을 찾기 위해 사용되는가?", "1. 최단 경로", "2.
최대값", "3. 중간 값", "4. 두 점 사이의 거리", 3},
        {"getchar() 함수의 헤더 파일은 무엇인가?", "1. <stdoi.h>", "2. <conio.h>", "3. <string.h>",
"4. <stdio.h>", 4},
        {"프로세스(Process)는 무엇을 나타내는가?", "1. 실제 파일", "2. 메모리 위치", "3. 실행 중인
프로그램", "4. 논리적 블록", 3}
};

```

```

int main()
{
    printf("[컴퓨터공학과, 김민철, 20230822]\n");
    printf("\n");

    int select, i;
    int correctCount = 0;
    int numQuestions = sizeof(bank) / sizeof(bank[0]);
    int* usedQuestions = (int*)malloc(numQuestions * sizeof(int));
    for (i = 0; i < 12; i++)
    {
        usedQuestions[i] = 0;
    }

    srand((unsigned)time(NULL));

    for (i = 0; i < 12; i++)
    {
        int randomIndex;
        do
        {
            randomIndex = rand() % 12;
        } while (usedQuestions[randomIndex]);

        usedQuestions[randomIndex] = 1;

        printf("%s\n", bank[randomIndex].question);
        printf("%s  ", bank[randomIndex].item1);
        printf("%s  ", bank[randomIndex].item2);
    }
}

```

```

printf("%s  ", bank[randomIndex].item3);
printf("%s  ", bank[randomIndex].item4);
while (1)
{
    printf("\n\n  답을 선택하시오(1~4): ");
    scanf("%d", &select);
    if (select < 1 || select > 4)
    {
        printf("1부터 4사이의 값을 입력하시오.\n");
    }
    else
    {
        break;
    }
}

if (select == bank[randomIndex].solution)
{
    printf("맞았습니다.\n\n");
    correctCount++;
}
else
{
    printf("틀렸습니다.\n\n");
}
}

printf("맞춘 문제 수: %d / %d\n", correctCount, 12);

free(usedQuestions);

return 0;
}

```

//실행 결과

[컴퓨터공학과, 김민철, 20230822]

SQL에서 'SELECT'는 무엇을 수행하는데 사용되는가?

1. 데이터 삽입 2. 데이터 삭제 3. 데이터 조회 4. 데이터 업데이트 // 답을 선택하시오(1~4): 3
맞았습니다.

HTML, CSS 및 JavaScript 중 어떤 언어가 웹 페이지의 디자인을 제어하는 데 사용되는가?

1. HTML 2. CSS 3. JavaScript 4. 모두 // 답을 선택하시오(1~4): 2
맞았습니다.

임베디드 장치에 가장 적합한 프로그래밍 언어는?

1. Python 2. Java 3. C 4. JavaScript // 답을 선택하시오(1~4): 3
맞았습니다.

getchar() 함수의 헤더 파일은 무엇인가?

1. <stdio.h> 2. <conio.h> 3. <string.h> 4. <stdlib.h> // 답을 선택하시오(1~4): 4
맞았습니다.

Big O 표기법에서 'O(n)' 무엇을 나타내는가?

1. 선형 시간 복잡도 2. 로그 시간 복잡도 3. 지수 시간 복잡도 4. 상수 시간 복잡도 // 답을 선택하시오(1~4): 2
틀렸습니다.

이진 검색(Binary Search) 알고리즘은 무엇을 찾기 위해 사용되는가?

1. 최단 경로 2. 최대값 3. 중간 값 4. 두 점 사이의 거리 // 답을 선택하시오(1~4): 3
맞았습니다.

데이터베이스에서 'SQL'은 무엇을 나타내는가?

1. 표준 쿼리 언어 2. 문산 파일 시스템 3. 그래픽 디자인 언어 4. 소프트웨어 개발 도구 // 답을 선택하시오(1~4): 1
맞았습니다.

TCP와 UDP 프로토콜 중 어떤 프로토콜이 연결 지향적이고 안정성을 제공하는가?

1. TCP 2. UDP 3. HTTP 4. IP // 답을 선택하시오(1~4): 1
맞았습니다.

C++에서 'cout'과 'cin' 무엇을 위해 사용되는가?

1. 문자열 처리 2. 파일 입출력 3. 표준 입출력 4. 그래픽 처리 // 답을 선택하시오(1~4): 1
틀렸습니다.

운영 체제의 주요 역할 중 하나는 무엇인가?

1. 웹 개발 2. 파일 관리 3. 음악 재생 4. 게임 실행 // 답을 선택하시오(1~4): 2
맞았습니다.

프로세스(Process)는 무엇을 나타내는가?

1. 실제 파일 2. 메모리 위치 3. 실행 중인 프로그램 4. 논리적 블록 // 답을 선택하시오(1~4): 3
맞았습니다.

서로 다른 자료형을 모을 수 있는 구조는?

1. 배열 2. 변수 3. 구조체 4. 포인터 // 답을 선택하시오(1~4): 3
맞았습니다.

맞춘 문제 수: 10 / 12

D:\Microsoft Visual Studio\Project27\#\x64\Debug\Project27.exe(프로세스 31600)이(가) 종료되었습니다(코드: 0x0).
이 창을 닫으려면 아무 키나 누르세요...

//P.585 - 01번

//프로그램 설계 사항

- 1) 'enum Genre' 열거형을 사용하여 도서의 장르를 정의합니다. 각 장르는 숫자로 표현됩니다.
- 2) 'struct Book' 구조체를 정의하여 도서 정보를 저장할 수 있는 데이터 구조를 생성합니다. 이 구조체에는 도서 제목, 저자 및 장르가 저장됩니다.
- 3) 'main' 함수가 실행될 때, 초기화된 'library' 배열과 다른 변수들을 선언합니다. 'library' 배열에는 도서 정보가 저장됩니다.
- 4) 'for' 반복문을 사용하여 최대 4권의 도서 정보를 입력합니다. 사용자로부터 도서 제목, 저자, 그리고 도서의 장르를 입력받습니다. 장르는 'enum Genre'에서 정의한 숫자 중 하나로 선택됩니다.
- 5) 사용자가 -1을 입력하면 입력이 종료되고, 그렇지 않으면 유효한 장르가 입력될 때까지 계속 입력을 받습니다.
- 6) 입력이 종료되면, 입력된 도서 정보를 도서 목록으로 출력합니다. 각 도서의 제목, 저자, 및 장르는 사용자가 입력한 정보와 함께 출력됩니다.

//코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#define MAX_BOOK 4

enum Genre {
    COMIC,
    SF,
    DOCU,
    DRAMA
};

struct Book {
    char title[100];
    char author[100];
    enum Genre genre;
};

int main()
{
    printf("[컴퓨터공학과, 김민철, 20230822]\n");
    printf("\n");

    struct Book library[MAX_BOOK];
    int index = 0;
```

```

for (int i = 0; i < MAX_BOOK; i++)
{
    printf("도서 제목: ");
    scanf(" %[^\\n]", library[index].title);

    printf("저자: ");
    scanf(" %[^\\n]", library[index].author);

    int genreChoice;
    printf("장르 (0: COMIC, 1: SF, 2: DOCU, 3: DRAMA)(-1을 입력하면 종료): ");
    scanf("%d", &genreChoice);

    printf("-----\\n");

    if (genreChoice == -1)
    {
        break;
    }

    if (genreChoice >= 0 && genreChoice <= 3)
    {
        library[index].genre = (enum Genre)genreChoice;
        index++;
    }
    else
    {
        printf("유효한 장르를 선택해주세요 (0에서 3 사이).\\n");
    }
}

printf("\\n도서 목록\\n");

for (int i = 0; i < index; i++)
{
    printf("도서 제목: %s\\n", library[i].title);
    printf("저자: %s\\n", library[i].author);

    const char* genreStr = NULL;
    switch (library[i].genre)
    {
    case COMIC:
        genreStr = "COMIC";
        break;

```



```

        case SF:
            genreStr = "SF";
            break;
        case DOCU:
            genreStr = "DOCU";
            break;
        case DRAMA:
            genreStr = "DRAMA";
            break;
        default:
            genreStr = "알 수 없음";
    }
    printf("장르: %s\n\n", genreStr);
}

return 0;
}

```

//실행 결과

[컴퓨터공학과, 김민철, 20230822]

도서 제목: 바람과 함께 사라지다
 저자: 마가렛 미첼
 장르 (0: COMIC, 1: SF, 2: DOCU, 3: DRAMA)(-1을 입력하면 종료): 3

 도서 제목: 토끼와 거북이
 저자: 푸른숲글방
 장르 (0: COMIC, 1: SF, 2: DOCU, 3: DRAMA)(-1을 입력하면 종료): 0

 도서 제목: 마블 코믹스
 저자: 마블
 장르 (0: COMIC, 1: SF, 2: DOCU, 3: DRAMA)(-1을 입력하면 종료): 0

 도서 제목: 심청전
 저자: 알 수 없음
 장르 (0: COMIC, 1: SF, 2: DOCU, 3: DRAMA)(-1을 입력하면 종료): 2

 도서 목록
 도서 제목: 바람과 함께 사라지다
 저자: 마가렛 미첼
 장르: DRAMA

도서 제목: 토끼와 거북이
 저자: 푸른숲글방
 장르: COMIC

도서 제목: 마블 코믹스
 저자: 마블
 장르: COMIC

도서 제목: 심청전
 저자: 알 수 없음
 장르: DOCU

C:\Users\user\source\repos\Project165\x64\Debug\Project165.exe(프로세스 19272개)이(가) 종료되었습니다(코드: 0개).
 이 창을 닫으려면 아무 키나 누르세요...

//P.585 - 02번

//프로그램 설계 사항

- 1) 'enum Genre' 열거형을 사용하여 도서의 장르를 정의합니다. 각 장르는 숫자로 표현됩니다.
- 2) 'struct Book' 구조체를 정의하여 도서 정보를 저장할 수 있는 데이터 구조를 생성합니다. 이 구조체에는 도서 제목, 저자 및 장르가 저장됩니다.
- 3) 'equal_author' 함수는 두 도서 구조체를 받아서 두 도서의 저자를 비교하여 같을 경우 1을 반환하고, 다를 경우 0을 반환합니다.
- 4) 'main' 함수가 실행될 때, 초기화된 두 개의 'struct Book' 변수인 'book1'과 'book2'를 선언합니다.
- 5) 사용자로부터 첫 번째 도서 정보를 입력받습니다. 도서 제목, 저자, 및 장르를 입력합니다.
- 6) 사용자로부터 두 번째 도서 정보를 입력받습니다. 역시 도서 제목, 저자, 및 장르를 입력합니다.
- 7) 'equal_author' 함수를 호출하여 두 도서의 저자가 같은지 확인합니다.
- 8) 'result' 변수에 저장된 결과에 따라 두 도서의 저자가 같은지, 다른지를 출력합니다.

//코드

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

enum Genre
{
    COMIC,
    SF,
    DOCU,
    DRAMA
};

struct Book
{
    char title[100];
    char author[100];
    enum Genre genre;
};

int equal_author(struct Book b1, struct Book b2)
{
    return strcmp(b1.author, b2.author) == 0;
}

int main()
{
    printf("[컴퓨터공학과, 김민철, 20230822]\n");
```

```

printf("\n");

struct Book book1, book2;

printf("도서 1 정보 입력\n");
printf("도서 제목: ");
scanf(" %[^\\n]", book1.title);
printf("저자: ");
scanf(" %[^\\n]", book1.author);
int genreChoice;
printf("장르 (0: COMIC, 1: SF, 2: DOCU, 3: DRAMA): ");
scanf("%d", &genreChoice);
book1.genre = (enum Genre)genreChoice;

printf("\n도서 2 정보 입력\n");
printf("도서 제목: ");
scanf(" %[^\\n]", book2.title);
printf("저자: ");
scanf(" %[^\\n]", book2.author);
printf("장르 (0: COMIC, 1: SF, 2: DOCU, 3: DRAMA): ");
scanf("%d", &genreChoice);
book2.genre = (enum Genre)genreChoice;

int result = equal_author(book1, book2);

if (result == 1)
{
    printf("\n두 책의 저자가 같습니다.\n");
}
else
{
    printf("\n두 책의 저자가 다릅니다.\n");
}

return 0;
}

```

//실행 결과

[컴퓨터공학과, 김민철, 20230822]

도서 1 정보 입력
도서 제목: 바람과 함께 사라지다
저자: 마가렛 미첼
장르 (0: COMIC, 1: SF, 2: DOCU, 3: DRAMA): 3

도서 2 정보 입력
도서 제목: 노인과 바다
저자: 헤밍웨이
장르 (0: COMIC, 1: SF, 2: DOCU, 3: DRAMA): 3

두 책의 저자가 다릅니다.

C:\Users\user\source\repos\Project166\x64\Debug\Project166.exe(프로세스 20708개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...|

//P.589 - 14번

//프로그램 설계 사항

- 1) 'struct song' struct song 구조체를 사용하여 곡의 정보를 저장합니다. 이 구조체에는 곡의 제목, 가수, 위치, 그리고 장르를 나타내는 숫자가 저장됩니다.
- 2) 'struct song s[1000]' 배열을 선언하여 최대 1000개의 음악 곡을 저장할 수 있는 공간을 확보합니다.
- 3) 'main' 함수에서는 사용자로부터 입력을 받고, 입력에 따라 다음과 같은 작업을 수행합니다:
 - 3-1) '1'을 선택하면 사용자로부터 새로운 곡의 정보를 입력받아 배열에 추가합니다.
 - 3-2) '2'를 선택하면 곡의 제목과 가수를 입력하고, 곡의 정보를 검색하여 출력합니다.
 - 3-3) '3'을 선택하면 곡의 제목과 가수를 입력하고, 해당 곡을 삭제할 것인지 물어봅니다.'y'를 선택하면 해당 곡을 삭제하고, 'n'을 선택하면 삭제를 취소합니다.
 - 3-4) '4'를 선택하면 프로그램을 종료합니다.
- 4) 'add_num' 변수는 현재 배열에 저장된 곡의 수를 추적합니다. 이 변수를 사용하여 배열에 새로운 곡을 추가하고, 삭제 시에도 업데이트합니다.
- 5) 'search_t'와 'search_n' 변수는 사용자가 검색하려는 곡의 제목과 가수를 입력받습니다.
- 6) 프로그램은 'while' 루프를 사용하여 사용자의 입력을 계속해서 받아들입니다. 선택된 메뉴에 따라서 각각의 작업을 수행하며, 유효하지 않은 입력에 대해서 적절한 에러 메시지를 출력합니다.

//코드

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

struct song
{
    char title[100];
    char name[100];
    char location[100];
    int genre;
};

struct song s[1000] = { NULL, };

int main(void)
{
    printf("[컴퓨터공학과, 김민철, 20230822]\n");
    printf("\n");

    int add_num = 0;
    int k = 0;
    int num = 0;
    int i = 0;
```

```

int j = 0;
int c = 0;
char search_t[1000] = { NULL, };
char search_n[1000] = { NULL, };
char copy[1000] = { NULL, };
char NUL[1000] = { NULL, };
char answer = { NULL };
while (1)
{
    printf("=====\\n  1.  추가\\n  2.  출력\\n  3.  삭제\\n  4.  종료
\\n=====\\n");
    printf("정수값을 입력하시오 : ");
    scanf("%d", &num);
    if (num == 1)
    {
        if (add_num >= 1000)
        {
            printf("곡을 더 저장할 용량이 부족합니다. 곡을 추가하려면 다른 곡을 삭제해주
세요.\\n");
            i = 0;
        }
        else
        {
            while ((c = getchar()) != '\\n' && c != EOF);
            printf("제목:");
            scanf("%[^\\n]s", s[add_num].title);
            while ((c = getchar()) != '\\n' && c != EOF);
            printf("가수:");
            scanf("%[^\\n]s", s[add_num].name);
            while ((c = getchar()) != '\\n' && c != EOF);
            printf("위치:");
            scanf("%[^\\n]s", s[add_num].location);
            while ((c = getchar()) != '\\n' && c != EOF);
            printf("장르(0: 가요 ,1: 팝, 2: 클래식, 3: 영화음악)");
            scanf("%d", &s[add_num].genre);
            if (s[add_num].genre > 3 || s[add_num].genre < 0)
            {
                printf("다시 입력해주세요.\\n");
                printf("장르(0: 가요 ,1: 팝, 2: 클래식, 3: 영화음악)");
                scanf("%d", &s[add_num].genre);
            }
            add_num++;
        }
    }
}

```

```

    }
    else if (num == 2)
    {
        printf("곡의 제목과 가수를 입력해주세요.\n");
        while ((c = getchar()) != '\n' && c != EOF);
        printf("제목:");
        scanf("%[^\n]s", search_t);
        while ((c = getchar()) != '\n' && c != EOF);
        printf("가수:");
        scanf("%[^\n]s", search_n);
        while (1)
        {
            if (strcmp(s[i].title, search_t) == 0 && strcmp(s[i].name, search_n) == 0)
            {
                printf("%s의 %s는 %s에 위치하고 있으며 ", s[i].name, s[i].title,
s[i].location);

                if (s[i].genre == 0)
                    printf("장르는 가요입니다.\n");
                else if (s[i].genre == 1)
                    printf("장르는 팝입니다.\n");
                else if (s[i].genre == 2)
                    printf("장르는 클래식입니다.\n");
                else if (s[i].genre == 3)
                    printf("장르는 영화음악입니다.\n");
                while ((c = getchar()) != '\n' && c != EOF);
                break;
            }
            if (s[i].title == NULL || i >= 1000)
            {
                printf("곡을 찾지 못했습니다.\n");
                break;
            }
            i++;
        }
        i = 0;
    }
    else if (num == 3)
    {
        printf("곡의 제목과 가수를 입력해주세요.\n");
        while ((c = getchar()) != '\n' && c != EOF);
        printf("제목:");
        scanf("%[^\n]s", search_t);
        while ((c = getchar()) != '\n' && c != EOF);
    }
}

```

```

printf("가수:");
scanf("%[^\\n]s", search_n);
while (1)
{
    if (strcmp(s[i].title, search_t) == 0 && strcmp(s[i].name, search_n) == 0)
    {
        printf("%s의 %s이며 %s에 위치하고 있는 곡이 맞습니까? (y or n)",
s[i].name, s[i].title, s[i].location);
        while ((c = getchar()) != '\\n' && c != EOF);
        scanf("%c", &answer);
        if (answer == 'y')
        {
            printf("곡을 삭제합니다.\\n");
            j = i;
            while (1)
            {
                strcpy(s[j].title, s[j + 1].title);
                strcpy(s[j].name, s[j + 1].name);
                strcpy(s[j].location, s[j + 1].location);
                s[j].genre = s[j + 1].genre;
                if (strcmp(s[j + 1].title, NUL) == 0)
                    break;
                j++;
            }
            while ((c = getchar()) != '\\n' && c != EOF);
            strcpy(copy, NUL);
            answer = 0;
            i = 0;
            j = 0;
            add_num--;
            break;
        }
        else
        {
            printf("삭제가 취소되었습니다.\\n");
            i = 0;
            break;
        }
    }
}
if (s[i].title == NULL || i >= 1000)
{
    printf("입력이 잘못되었습니다.\\n");
    break;
}

```



```

        }
        i++;
    }
}
else if (num == 4)
{
    printf("프로그램을 종료합니다.\n");
    break;
}
else
    printf("다시 입력하세요.\n");
}

return 0;
}

```

//실행 결과

[컴퓨터공학과, 김민철, 20230822]

```

=====
1. 추가
2. 출력
3. 삭제
4. 종료
=====

```

```

정수값을 입력하시오 : 1
제목:Top of The World
가수:Carpenters
위치:c:\mp3\top.mp3
장르(0: 가요 ,1: 팝, 2: 클래식, 3: 영화음악)1
=====

```

```

1. 추가
2. 출력
3. 삭제
4. 종료
=====

```

```

정수값을 입력하시오 : 2
곡의 제목과 가수를 입력해주세요.
제목:Top of The World
가수:Carpenters
Carpenters의 Top of The World는 c:\mp3\top.mp3에 위치하고 있으며 장르는 팝입니다.
=====

```

```

1. 추가
2. 출력
3. 삭제
4. 종료
=====

```

```

정수값을 입력하시오 : 3
곡의 제목과 가수를 입력해주세요.
제목:Top of The World
가수:Carpenters
Carpenters의 Top of The World이며 c:\mp3\top.mp3에 위치하고 있는 곡이 맞습니까? (y or n)y
곡을 삭제합니다.
=====

```

```

1. 추가
2. 출력
3. 삭제
4. 종료
=====

```

```

정수값을 입력하시오 : 4
프로그램을 종료합니다.

```

C:\Users\user\source\repos\Project162\x64\Debug\Project162.exe(프로세스 17080개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

//P.589 - 15번

//프로그램 설계 사항

- 1) 'struct Book' 구조체는 도서의 정보를 저장하기 위한 구조체로, 'int bookNumber'로 도서 번호, 'char title[100]'로 도서 제목, 'char author[100]'로 도서 저자, 그리고 'int isAvailable'로 도서의 대출 가능 여부를 저장합니다.
- 2) 'struct Book library[MAX_BOOKS]' 배열은 최대 100권의 도서 정보를 저장할 수 있는 배열로, 'bookCount' 변수로 현재 저장된 도서의 개수를 추적합니다.
- 3) 프로그램은 main 함수를 사용하여 사용자와 상호 작용하며 다음 메뉴를 제공합니다:
 - 도서 번호로 책 찾기
 - 저자 이름으로 책 찾기
 - 제목으로 책 찾기
 - 새로운 책 추가
 - 도서관이 소장한 도서의 수 표시
- 4) 사용자가 메뉴를 선택하면, 해당 기능을 수행하는 함수가 호출됩니다.
 - 'findBookByNumber()': 도서 번호를 입력받아 해당 번호의 도서 정보를 출력합니다.
 - 'findBookByAuthor()': 저자 이름을 입력받아 해당 저자의 도서 정보를 출력합니다.
 - 'findBookByTitle()': 도서 제목을 입력받아 해당 제목의 도서 정보를 출력합니다.
 - 'addNewBook()': 새로운 도서의 정보를 입력받아 도서 목록에 추가합니다.
 - 'displayLibrarySize()': 도서관이 소장한 도서의 수를 출력합니다.

//코드

```
#define _CRT_SECURE_NO_WARNINGS
```

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

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

```
#define MAX_BOOKS 100
```

```
struct Book{  
    int bookNumber;  
    char title[100];  
    char author[100];  
    int isAvailable;  
};
```

```
struct Book library[MAX_BOOKS];
```

```
int bookCount = 0;
```

```
void findBookByNumber();
```

```
void findBookByAuthor();
```

```
void findBookByTitle();
```

```
void addNewBook();
```

```
void displayLibrarySize();
```

```

int main()
{
    printf("[컴퓨터공학과, 김민철, 20230822]\n");
    printf("\n");

    int choice = 0;

    while (choice != -1)
    {
        printf("===== \n");
        printf("1. 도서 번호로 책 찾기\n");
        printf("2. 저자 이름으로 책 찾기\n");
        printf("3. 제목으로 책 찾기\n");
        printf("4. 새로운 책 추가\n");
        printf("5. 도서관이 소장한 도서의 수 표시\n");
        printf("===== \n");
        printf("메뉴 중 하나를 선택하시오(-1을 입력하면 종료): ");
        scanf("%d", &choice);
        printf("\n");

        if (choice == -1)
        {
            break;
        }

        switch (choice)
        {
            case 1:
                findBookByNumber();
                break;
            case 2:
                findBookByAuthor();
                break;
            case 3:
                findBookByTitle();
                break;
            case 4:
                addNewBook();
                break;
            case 5:
                displayLibrarySize();
                break;
            default:

```

```

        printf("잘못된 선택입니다. 1부터 5사이의 값을 입력하세요.\n");
    }
    printf("\n");
}

return 0;
}

void findBookByNumber()
{
    int bookNumber;
    printf("도서 번호를 입력하십시오: ");
    scanf("%d", &bookNumber);

    int found = 0; // 해당 도서 번호의 책을 찾았는지 나타내기 위해 씬

    for (int i = 0; i < bookCount; i++)
    {
        if (library[i].bookNumber == bookNumber)
        {
            printf("도서 번호: %d\n", library[i].bookNumber);
            printf("제목: %s\n", library[i].title);
            printf("저자: %s\n", library[i].author);
            printf("대출 여부: %s\n", library[i].isAvailable ? "대출 가능" : "대출 중");
            found = 1;

            return;
        }
    }

    if (found == 0)
    {
        printf("해당 번호의 책을 찾을 수 없습니다.\n");
    }
}

void findBookByAuthor()
{
    char author[100];
    printf("저자 이름을 입력하십시오: ");
    scanf("%99[^\n]", author); // 띄어쓰기를 포함한 문자열을 입력하기 위해 %99[^\n] 사용

    int found = 0; // 해당 저자의 책을 찾았는지 나타내기 위해 씬

```

```

for (int i = 0; i < bookCount; i++)
{
    if (strcmp(library[i].author, author) == 0)
    {
        printf("-----\n");
        printf("도서 번호: %d\n", library[i].bookNumber);
        printf("제목: %s\n", library[i].title);
        printf("대출 여부: %s\n", library[i].isAvailable ? "대출 가능" : "대출 중");
        found = 1;
    }
}

if (found == 0)
{
    printf("해당 저자의 책을 찾을 수 없습니다.\n");
}
}

void findBookByTitle()
{
    char title[100];
    printf("제목을 입력하십시오: ");
    scanf(" %99[^\n]", title); // 띄어쓰기를 포함한 문자열을 입력하기 위해 %99[^\n] 사용

    int found = 0; // 해당 제목의 책을 찾았는지 나타내기 위해 씀

    for (int i = 0; i < bookCount; i++)
    {
        if (strcmp(library[i].title, title) == 0)
        {
            printf("도서 번호: %d\n", library[i].bookNumber);
            printf("저자: %s\n", library[i].author);
            printf("대출 여부: %s\n", library[i].isAvailable ? "대출 가능" : "대출 중");
            found = 1;
        }
    }

    if (found == 0)
    {
        printf("해당 제목의 책을 찾을 수 없습니다.\n");
    }
}

```

```

void addNewBook()
{
    if (bookCount >= MAX_BOOKS)
    {
        printf("더 이상 책을 추가할 수 없습니다. 도서관의 용량을 초과했습니다.\n");
        printf("\n");

        return;
    }

    struct Book newBook;
    newBook.bookNumber = bookCount + 1;
    printf("도서 제목을 입력하시오: ");
    scanf(" %[^\\n]", newBook.title); // 띄어쓰기를 포함한 문자열을 입력하기 위해 %[^\\n] 사용
    printf("도서 저자를 입력하시오: ");
    scanf(" %[^\\n]", newBook.author); // 띄어쓰기를 포함한 문자열을 입력하기 위해 %[^\\n]
    사용
    newBook.isAvailable = 1;

    library[bookCount] = newBook;
    bookCount++;

    printf("도서가 추가되었습니다.\n");
}

void displayLibrarySize()
{
    printf("도서관이 소장한 도서의 수: %d\\n", bookCount);
}

```

//실행 결과(밑으로 내리면 있습니다.)

[컴퓨터공학과, 김민철, 20230822]

- ```
=====
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 선택하시오(-1을 입력하면 종료): 4

도서 제목을 입력하시오: 파이썬 프로그래밍  
도서 저자를 입력하시오: 김상근  
도서가 추가되었습니다.

- ```
=====
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 선택하시오(-1을 입력하면 종료): 4

도서 제목을 입력하시오: C언어 EXPRESS
도서 저자를 입력하시오: 천인국
도서가 추가되었습니다.

- ```
=====
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 선택하시오(-1을 입력하면 종료): 1

도서 번호를 입력하십시오: 1  
도서 번호: 1  
제목: 파이썬 프로그래밍  
저자: 김상근  
대출 여부: 대출 가능

- ```
=====
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 선택하시오(-1을 입력하면 종료): 1

도서 번호를 입력하십시오: 4
해당 번호의 책을 찾을 수 없습니다.

- ```
=====
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 선택하시오(-1을 입력하면 종료): 2



저자 이름을 입력하십시오: 김상근

도서 번호: 1  
제목: 파이썬 프로그래밍  
대출 여부: 대출 가능

- ```
=====
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 선택하십시오(-1을 입력하면 종료): 2

저자 이름을 입력하십시오: 홍길동
해당 저자의 책을 찾을 수 없습니다.

- ```
=====
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 선택하십시오(-1을 입력하면 종료): 3

제목을 입력하십시오: 파이썬 프로그래밍  
도서 번호: 1  
저자: 김상근  
대출 여부: 대출 가능

- ```
=====
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 선택하십시오(-1을 입력하면 종료): 3

제목을 입력하십시오: 프로그래밍 파이썬
해당 제목의 책을 찾을 수 없습니다.

- ```
=====
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 선택하십시오(-1을 입력하면 종료): 5

도서관이 소장한 도서의 수: 2

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

```
=====
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 선택하시오(-1을 입력하면 종료): -1

D:\Microsoft Visual Studio\Project26\x64\Debug\Project26.exe(프로세스 10752개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...