60181641_박지수

1-1 구조체 포인터 배열 사용

코드

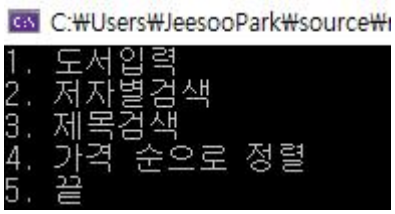
```
#include <stdio.h>
#include<stdlib.h>
#include<string.h>
typedef struct book {
	char title[40];
	char writer[20];
	int price;
}BOOK;
int compareByPrice(const void* b1, const void* b2);
int compareByWriter(const void* b1, const void* b2);
int compareByTitle(const void* b1, const void* b2);

int main(void) {
	BOOK* book[10];
	BOOK writer, title;
	BOOK* ptr;

	int counter = 0;
	int count = 10;

	for (int i = 0;i < sizeof(book) / sizeof(BOOK*);i++) {
		book[i] = malloc(sizeof(BOOK));
		memset(book[i], 0, sizeof(BOOK));
	}

	int select = 1;
	int found = -1;

	while (1) {
		printf("1. 도서입력\n");
		printf("2. 저자별검색\n");
		printf("3. 제목검색\n");
		printf("4. 가격 순으로 정렬\n");
		printf("5. 끝\n");
		scanf("%d", &select);
		switch (select)
		{
		case 1:
```

```
            if (counter == count) {
                printf("더이상의 책 정보를 입력할 수 없습니다.\n");
            }
            else {

                if (book[counter] == NULL) {
                    return -1;
                }
                while (getchar() != '\n');
                printf("저자 :");
                gets(book[counter]->writer);
                printf("책 제목 :");
                gets(book[counter]->title);
                printf("가 격 :");
                scanf("%d", &book[counter]->price);
                counter = counter + 1;
            }
            break;
        case 2:
            printf("저자명을 입력해주세요\n");
            getchar();

            ptr = &writer;
            gets(ptr->writer);
            qsort(book, counter, sizeof(BOOK*), compareByWriter);

            BOOK** result = (BOOK**)bsearch(&ptr, book, counter, sizeof(BOOK*),
compareByWriter);
            if (result != NULL)
            {
                printf("%s가 작성한책은 %s이며 가격은 %d입니다.\n", (*result)->writer,
(*result)->title, (*result)->price);
            }
            else
            {
                printf("해당 도서가 존재하지 않습니다.\n");
            }
            break;
        case 3:
            printf("도서명을 입력해주세요\n");
```

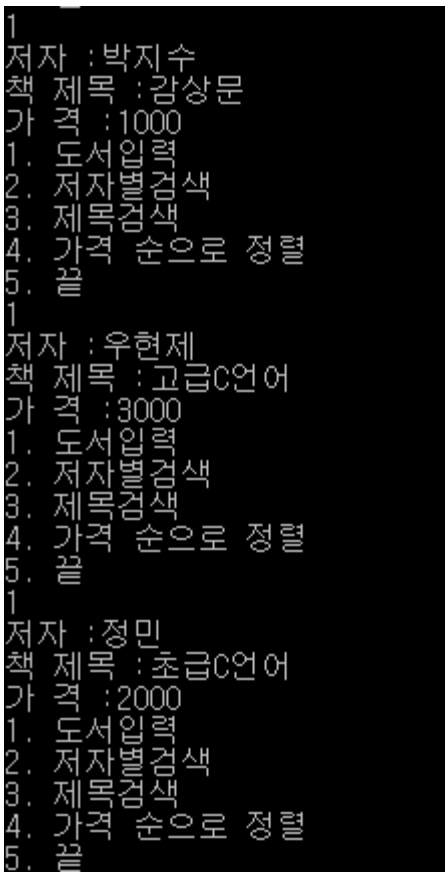
```
            getchar();

            ptr = &title;
            gets(ptr->title);
            qsort(book, counter, sizeof(BOOK*), compareByTitle);

            BOOK** result2 = (BOOK**)bsearch(&ptr, book, counter, sizeof(BOOK*),
compareByTitle);
            if (result2 != NULL)
            {
                printf("%s는 %s가 작성한 책이며 가격은 %d입니다.\n", (*result2)->title,
(*result2)->writer, (*result2)->price);
            }
            else
            {
                printf("해당 도서가 존재하지 않습니다.\n");
            }
            break;
        case 4:
            qsort(book, counter, sizeof(book[0]), compareByPrice);
            printf("정렬 후 가격:\n");
            for (int i = 0;i < counter;i++) {
                printf("%d\n", book[i]->price);
            }
            break;
        case 5:
            exit(0);
            break;
        default:
            printf("잘못된 입력입니다.");
            break;
        }

    }

    for (int i = 0; i < sizeof(book) / sizeof(BOOK*); i++)
    {
        free(book[i]);
        book[i] = NULL;
    }
```

```
    return 0;
}

int compareByTitle(const void* b1, const void* b2)
{
    const BOOK* B1 = *(BOOK**)b1;
    const BOOK* B2 = *(BOOK**)b2;
    return  strcmp(B1->title, B2->title);

}
int compareByWriter(const void* b1, const void* b2)
{
    const BOOK* B1 = *(BOOK**)b1;
    const BOOK* B2 = *(BOOK**)b2;
    return  strcmp(B1->writer, B2->writer);
}
int compareByPrice(const void* b1, const void* b2) {
    const BOOK* B1 = *(BOOK**)b1;
    const BOOK* B2 = *(BOOK**)b2;

    if ((B1->price < B2->price)) {
        return 1;
    }
    else if ((B1->price > B2->price)) {
        return -1;
    }
    else {
        return 0;
    }
}
```

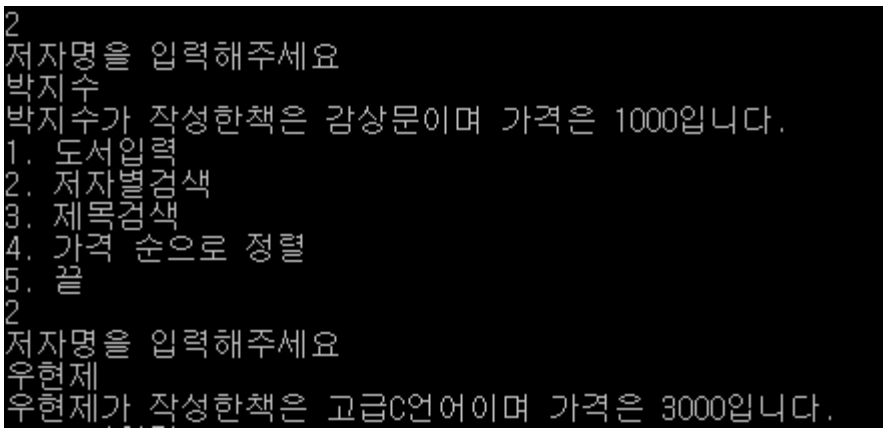
메뉴 출력



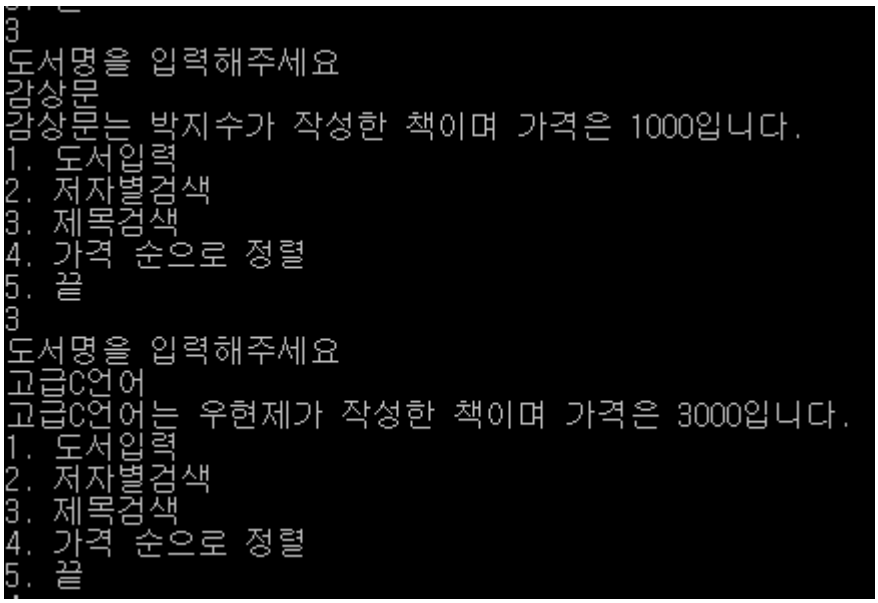
도서 정보 입력



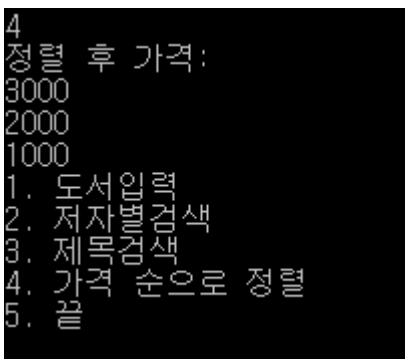
메뉴 2번



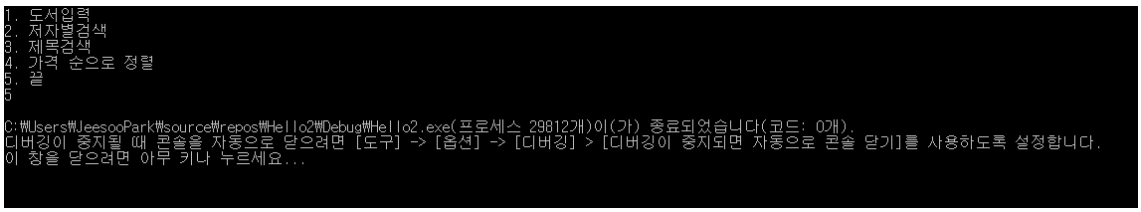
메뉴 3번



메뉴 4번



메뉴 5번 선택시 종료



2. 시작될 때 최대 저장 가능한 도서 정보의 개수를 입력받아서 동적 메모리에 book구조체 배열을 한꺼번에 할당하는 코드.

```
#include <stdio.h>
#include<stdlib.h>
#include<string.h>
typedef struct book {
        char title[40];
        char writer[20];
        int price;
```

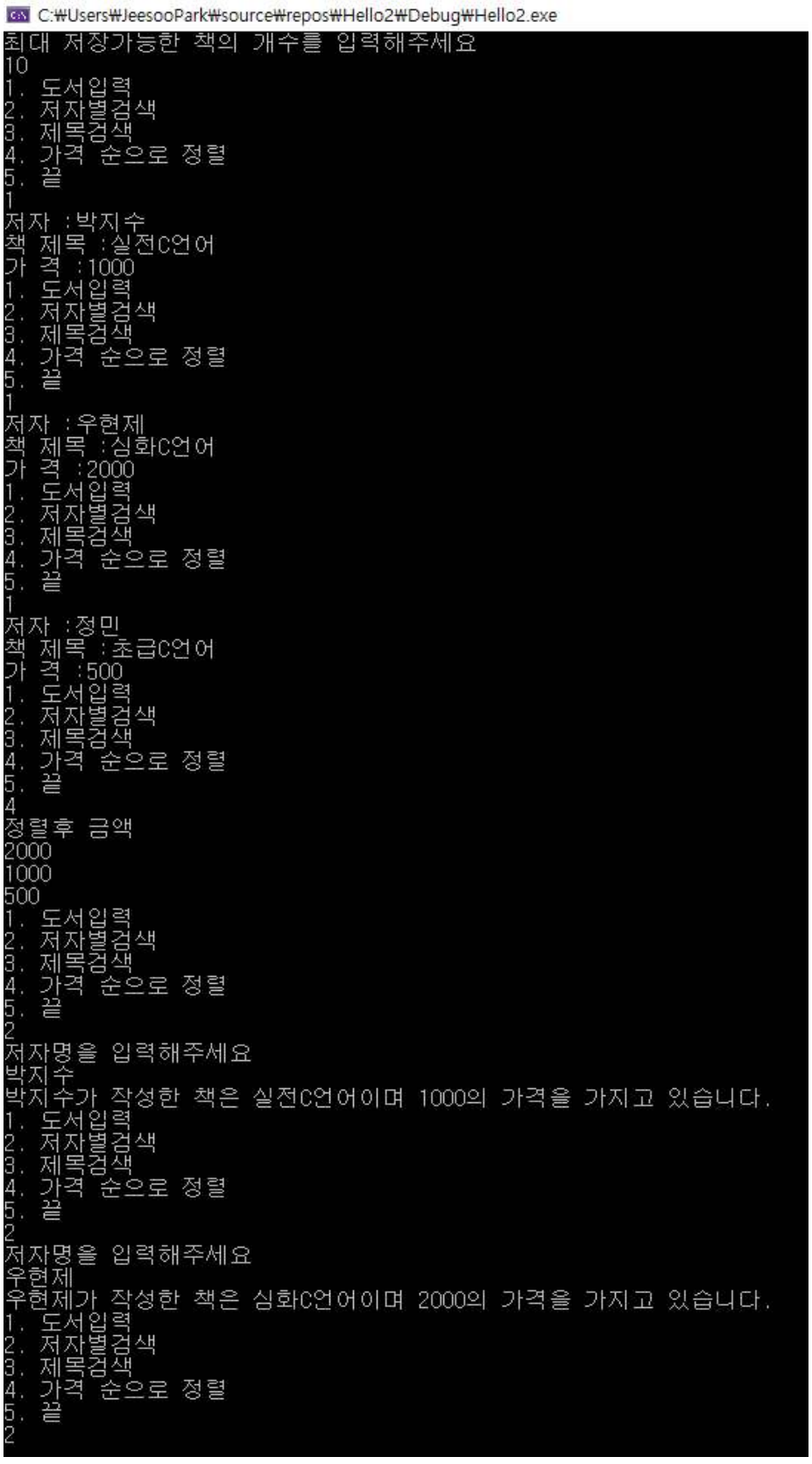
```
}BOOK;
BOOK* AllocateBooks(int count);
void InputBookInfo(BOOK* contacts, int count);
int compareByPrice(const void* b1, const void* b2);
int compareByWriter(const void* b1, const void* b2);
int compareByTitle(const void* b1, const void* b2);
int main(void) {
	BOOK* book = NULL;
	BOOK* result,writer, title;
	int count = 0;
	int counter = 0;
	printf("최대 저장가능한 책의 개수를 입력해주세요\n");
	scanf("%d", &count);
	while (getchar() != '\n');
	book = AllocateBooks(count);
	int select = 1;
	int found = -1;
	char key[40] = { 0 };
	while (1) {
		printf("1. 도서입력\n");
		printf("2. 저자별검색\n");
		printf("3. 제목검색\n");
		printf("4. 가격 순으로 정렬\n");
		printf("5. 끝\n");
		scanf("%d", &select);
		switch (select)
		{
		case 1:
			if (counter == count) {
				printf("더이상의 책 정보를 입력할 수 없습니다.\n");
			}
			else {

				if (book == NULL) {
					return -1;
				}
				while (getchar() != '\n');
				InputBookInfo(book, counter);
				counter = counter + 1;
			}
			break;
```

```
		case 2:
			printf("저자명을 입력해주세요\n");
			getchar();
			gets(writer.writer);
				qsort(book, count, sizeof(*book), compareByWriter);
				result = bsearch(&writer, book, count, sizeof(BOOK),
compareByWriter);
				if (result != NULL) {
					printf("%s가 작성한 책은 %s이며 %d의 가격을
가지고 있습니다.\n", result->writer, result->title, result->price);
				}
				else {
					printf("해당 도서가 존재하지 않습니다.\n");
				}
			break;
		case 3:
			printf("도서명을 입력해주세요\n");
			getchar();
			gets(title.title);
				qsort(book, count, sizeof(*book), compareByTitle);
				result = bsearch(&title, book, count, sizeof(BOOK),
compareByTitle);
				if (result != NULL) {
					printf("%s의 책은 %s가 작성하였고 %d의 가격
입니다.\n", result->title, result->writer, result->price);
				}
				else {
					printf("해당 도서가 존재하지 않습니다.\n");
				}
			break;
		case 4:
			qsort(book, count, sizeof(*book), compareByPrice);
			printf("정렬후 금액\n");
			for (int i = 0;i < counter;i++) {
				printf("%d\n", book[i].price);
			}
			break;
		case 5:
			free(book);
			book = NULL;
			exit(0);
```

```
                        break;
                default:
                        printf("잘못된 입력입니다.\n");
                        break;
                }
        }
        return 0;
}
int compareByWriter(const void* b1, const void* b2) {
        const BOOK* B1 = (const BOOK*)b1;
        const BOOK* B2 = (const BOOK*)b2;
        return strcmp(B1->writer, B2->writer);
}
int compareByTitle(const void* b1, const void* b2) {
        const BOOK* B1 = (const BOOK*)b1;
        const BOOK* B2 = (const BOOK*)b2;
        return strcmp(B1->title, B2->title);
}
int compareByPrice(const void* b1, const void* b2) {
        const BOOK* B1 = (const BOOK*)b1;
        const BOOK* B2 = (const BOOK*)b2;
        if (((B1->price == B2->price))) {
                return 0;
        }
        else if ((B1->price < B2->price)) {
                return 1;
        }
        else if ((B1->price > B2->price)) {
                return -1;
        }
}
BOOK* AllocateBooks(int count)
{
        BOOK* result = NULL;
        result = (BOOK*)malloc(sizeof(BOOK) * count);
        if (result == NULL)
                printf("동적 메모리 할당 실패\n");
        else
                memset(result, 0, sizeof(BOOK) * count);
        return result;
}
```

```
void InputBookInfo(BOOK* books, int count)
{
        printf("저자 :");
        gets(books[count].writer);
        printf("책 제목 :");
        gets(books[count].title);
        printf("가 격 :");
        scanf("%d", &books[count].price);
}
```

결과


C:₩Users₩JeesooPark₩source₩repos₩Hello2₩Debug₩Hello2.exe
최대 저장가능한 책의 개수를 입력해주세요
10
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
1
저자 :박지수
책 제목 :실전C언어
가 격 :1000
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
1
저자 :우현제
책 제목 :심화C언어
가 격 :2000
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
1
저자 :정민
책 제목 :초급C언어
가 격 :500
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
4
정렬후 금액
2000
1000
500
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
2
저자명을 입력해주세요
박지수
박지수가 작성한 책은 실전C언어이며 1000의 가격을 가지고 있습니다.
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
2
저자명을 입력해주세요
우현제
우현제가 작성한 책은 심화C언어이며 2000의 가격을 가지고 있습니다.
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
2

```
저자명을 입력해주세요
정민
정민가 작성한 책은 초급C언어이며 500의 가격을 가지고 있습니다.
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
3
도서명을 입력해주세요
초급C언어
초급C언어의 책은 정민가 작성하였고 500의 가격입니다.
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
3
도서명을 입력해주세요
심화c언어
해당 도서가 존재하지 않습니다.
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
3
도서명을 입력해주세요
심화C언어
심화C언어의 책은 우현제가 작성하였고 2000의 가격입니다.
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
3
도서명을 입력해주세요
실전C언어
실전C언어의 책은 박지수가 작성하였고 1000의 가격입니다.
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
```

```
1. 도서입력
2. 저자별검색
3. 제목검색
4. 가격 순으로 정렬
5. 끝
5

C:₩Users₩JeesooPark₩source₩repos₩Hello2₩Debug₩Hello2.exe(프로세스 16760개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```