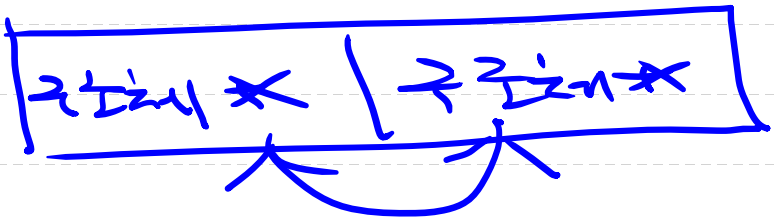
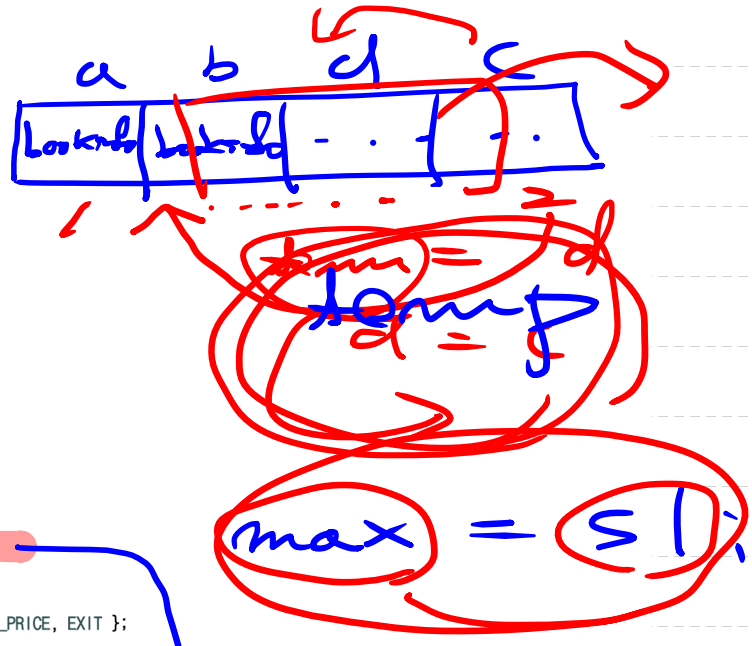
1. 성능

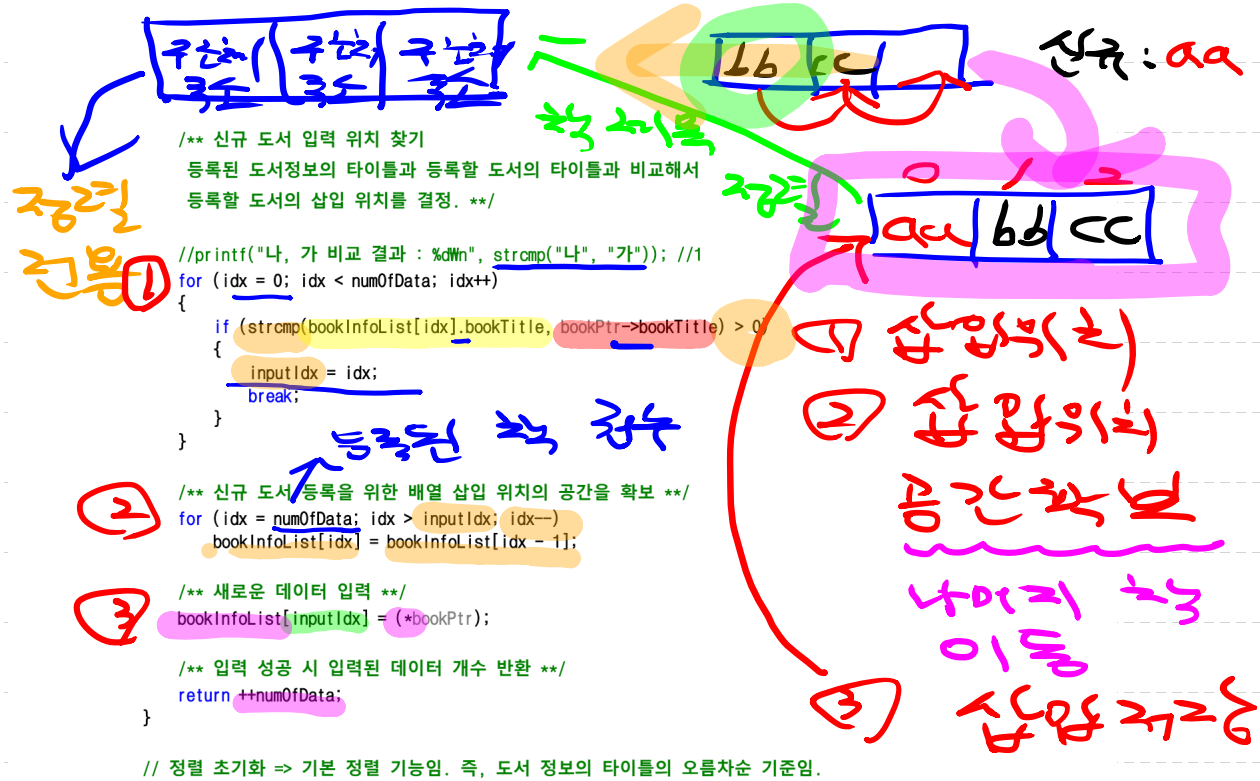
↳ 구조체 포인터 배열





```c
#include <stdio.h>
#include <string.h>
#define  STR_LEN     50
#define  BOOK_INFO_NUM   3

typedef struct bookInfo {
    char bookTitle[STR_LEN];
    char bookPub[STR_LEN];
    int bookPrice;
}BookInfo;

/** 배열에 저장된 책의 권수 **/
int numOfData = 0;

/** 도서정보를 저장할 배열 **/
BookInfo bookInfoList[BOOK_INFO_NUM];

/** 정렬 메뉴를 위한 상수 **/
enum { SORT_TITLE = 1, SORT_PUB, SORT_PRICE, EXIT };

/** 정렬후 결과를 저장할 매열 - 성능을 위해서 포인터 배열 사용 **/
BookInfo* bookSortList[BOOK_INFO_NUM];

int InsertList(BookInfo* bookPtr);

void InitSortList(void);
void PrintSortList(void);

void SortByTitle(void);
void SortByPubName(void);
void SortByPrice(void);

int main(void)
{
    int i, num;
    int choice;
    BookInfo book;

    printf("========= 도서 정보 입력 ========\n");
    for (i = 0; i < BOOK_INFO_NUM; i++)
    {
        printf("%d번째 도서 정보 입력. \n", (i + 1));

        printf("도서 제목: ");
        gets(book.bookTitle);

        printf("출판사 명: ");
        gets(book.bookPub);

        printf("도서 가격: ");
        scanf("%d", &(book.bookPrice));
        //fflush(stdin);
```

```
        getchar();

        num = InsertList(&book);
        if (num == -1)
            printf("입력에 실패하였습니다. \n");
        else
            printf("%d번째 입력 완료했습니다.\n\n", num);
    }

    InitSortList();
    while (1)
    {
        printf("======== 도서 정보 출력 ========\n");
        printf("정렬방식 메뉴입니다.\n");
        printf("1.도서 제목 순, 2.출판사 순, 3.가격 순, 4.종료\n");
        printf("정렬방식을 선택하세요. >> ");
        scanf("%d", &choice);
        if (choice == EXIT)
            break;

        switch (choice)
        {
        case SORT_TITLE:
            SortByTitle();
            PrintSortList();
            break;

        case SORT_PUB:
            SortByPubName();
            PrintSortList();
            break;

        case SORT_PRICE:
            SortByPrice();
            PrintSortList();
            break;
        }
    }

    return 0;
}

int InsertList(BookInfo* bookPtr)
{
    //도서 관리 배열에서 탐색 위치
    int idx;

    //신규 데이터가 등록될 배열의 위치
    int inputIdx = numOfData;

    if (numOfData >= BOOK_INFO_NUM)
        return -1; // -1은 입력 실패를 의미
```



```
/** 신규 도서 입력 위치 찾기
 등록된 도서정보의 타이틀과 등록할 도서의 타이틀과 비교해서
 등록할 도서의 삽입 위치를 결정. **/

//printf("나, 가 비교 결과 : %dWn", strcmp("나", "가")); //1
for (idx = 0; idx < numOfData; idx++)
{
    if (strcmp(bookInfoList[idx].bookTitle, bookPtr->bookTitle) > 0)
    {
        inputIdx = idx;
        break;
    }
}

/** 신규 도서 등록을 위한 배열 삽입 위치의 공간을 확보 **/
for (idx = numOfData; idx > inputIdx; idx--)
    bookInfoList[idx] = bookInfoList[idx - 1];

/** 새로운 데이터 입력 **/
bookInfoList[inputIdx] = (*bookPtr);

/** 입력 성공 시 입력된 데이터 개수 반환 **/
return ++numOfData;
}

// 정렬 초기화 => 기본 정렬 기능임. 즉, 도서 정보의 타이틀의 오름차순 기준임.
void InitSortList(void)
{
int i;
for (i = 0; i < BOOK_INFO_NUM; i++)
    bookSortList[i] = &bookInfoList[i];
}

void PrintSortList(void)
{
int i;
for (i = 0; i < BOOK_INFO_NUM; i++)
{
    printf("%d번째 도서 정보 출력. Wn", (i + 1));
    printf("도서 제목: %s Wn", bookSortList[i]->bookTitle);
    printf("출판사 명: %s Wn", bookSortList[i]->bookPub);
    printf("도서 가격: %d WnWn", bookSortList[i]->bookPrice);
}
}

void SortByTitle(void)
{
InitSortList();
}
```

```
void SortByPubName(void)
{
    int i, j;
    BookInfo* temp;

    for (i = 0; i < BOOK_INFO_NUM - 1; i++)
    {
        for (j = 0; j < (BOOK_INFO_NUM - i) - 1; j++)
        {
            if (strcmp(bookSortList[j]->bookPub, bookSortList[j + 1]->bookPub) > 0)
            {
                temp = bookSortList[j];
                bookSortList[j] = bookSortList[j + 1];
                bookSortList[j + 1] = temp;
            }
        }
    }
}

void SortByPrice(void)
{
    int i, j;
    BookInfo* temp;

    for (i = 0; i < BOOK_INFO_NUM - 1; i++)
    {
        for (j = 0; j < (BOOK_INFO_NUM - i) - 1; j++)
        {
            if (bookSortList[j]->bookPrice > bookSortList[j + 1]->bookPrice)
            {
                temp = bookSortList[j];
                bookSortList[j] = bookSortList[j + 1];
                bookSortList[j + 1] = temp;
            }
        }
    }
}
```