

---

# Midterm exam

제목 : 도서관리 프로그램

제출일 : 2020 년 6 월 05 일

이름 : 박용훈

학번 : 201901586

---



## 1. 프로그램 설계

2 차원 배열 사용

전역변수 사용

- 1) ADD = 단순 추가, 단 ID 에서 중복, 음수 체크
- 2) UPDATE = 인덱스를 입력받은 후에 수정할 정보를 추가로 받은 후 전체 수정
- 3) DELETE = 단순 삭제, STRCPY 를 사용해서 인덱스를 밀어서 덮어씌움
- 4) SEARCH = ID 를 입력하면 INDEX RETURN
- 5) DISPLAY = 저자, 출판사, ALL 순으로 설정

## 2. 함수 설명

### 1) MAIN 함수

WHILE 문과 SWITCH 문을 이용해서 사용자가 EXIT 하기 전까지는 프로그램이 종료되지 않게 설정했으며, 도서 정보를 모두 전역변수로 설정했기에 진입하는 함수는 모두 VOID 로 설정이 됨. 즉 따로 매개변수를 전달할 필요가 없음.

### 2) ADD 함수

도서 정보를 추가하는 함수이며, 도서 ID 를 입력하면 BOOL 변수를 CHECK 함수로 보내면서 중복상태를 RETURN 받아 ID 가 중복인지, 음수인지 확인을 하고, 중복이거나 음수라면 무한루프를 통해 올바른 값을 입력할 때 까지 입력받게함.

이후 나머지 값을 순서대로 입력받게함 그리고 COUNT 를 증가시켜 스택을 쌓아올림.

```
void ADD(void) { // 도서정보를 추가하는 함수.
    printf("Enter the Book ID : "); // 도서 ID 입력
    scanf("%d", &Book_ID[count]); // ID 를 Book_ID 글로벌 변수에 count 위치에 저장

    // BOOK ID 가 중복되거나 음수인경우를 이곳에서 걸러냄.
    int bool = check(Book_ID[count]); // Check 함수를 통해 bool 의 상태를 저장해놓음.
    if (bool == 1 || Book_ID[count] < 0) { // bool = 1 은 중복된다는 뜻, Book_ID[count] < 0 은 음수값을 의미
        // 둘중 하나라도 걸리면 if 문에 진입
        int bool2;
        while (1) { // 올바른 값을 입력할 때 까지 무한루프
            printf("The ID already exists or you entered a negative number.\n");
            printf("Enter the Book ID : ");
            scanf("%d", &Book_ID[count]);
            bool2 = check(Book_ID[count]);
            printf("%d\n", bool2);
            if (bool2 == 0 && Book_ID[count] >= 0) { // 만약 bool = 0 이 나와서 중복이 안되고 >=으로 0 이상의
```

수라면 break 를 통해 반복문 탈출.

```
        break;
    }
}
}

// 나머지 값들 차례대로 입력.
printf("Enter the Book Name : ");
scanf("%s", Book_Name[count]);

printf("Enter the date of publication of a book : ");
scanf("%s", Book_date_of_issue[count]);

printf("Enter the Publisher Name : ");
scanf("%s", Publisher_Name[count]);

printf("Enter the Publisher Phone Number : ");
scanf("%s", Publisher_Number[count]);

printf("Enter the Author Name : ");
scanf("%s", Author_Name[count]);

printf("Enter the Author e-mail address : ");
scanf("%s", Author_eaddress[count]);

count++; // 함수의 마지막에 count 를 증가시켜 스택 증가
}
```

### 3) UPDATE 함수

수정하고 싶은 도서정보의 인덱스를 입력받아서 그 도서를 수정함

아래는 거의 ADD 함수와 동일하며 단 위치를 COUNT 가 아닌 입력받은 INDEX 을  
기준으로 수정함으로써 추가가 아닌 수정이 되도록 했음.

```
void UPDATE(void) { // 업데이트 함수.
    int index; // 인덱스

    printf("Please enter the index of the book you want to modify : "); // 인덱스를 입력받으면 그 위치의
    정보들을 전부 새로 입력
    scanf("%d", &index); // 여기서 index 는 count 랑 비슷하다고 보면 됨.

    printf("Book ID : ");
    scanf("%d", &Book_ID[index]); // 여기서부터는 ADD 함수랑 똑같지만 보면 count 부분이 index 인 것을 알
    수 있음. ex) Book_ID[index]

    int bool = check(Book_ID[count]);
    if (bool == 1 || Book_ID[count] < 0) {
        int bool2;
        while (1) {
            printf("The ID already exists or you entered a negative number.\n");
            printf("Enter the Book ID : ");
            scanf("%d", &Book_ID[count]);
            bool2 = check(Book_ID[count]);
        }
    }
}
```

```

printf("%d\n", bool2);
if (bool2 == 0 && Book_ID[count] >= 0) {
    break;
}
}
}

printf("Book Name : ");
scanf("%s", Book_Name[index]);

printf("Book date of publish : ");
scanf("%s", Book_date_of_issue[index]);

printf("Publisher Name : ");
scanf("%s", Publisher_Name[index]);

printf("Publisher Phone number : ");
scanf("%s", Publisher_Number[index]);

printf("Author Name : ");
scanf("%s", Author_Name[index]);

printf("Author E-mail Address : ");
scanf("%s", Author_eaddress[index]);
}

```

#### 4) DELETE 함수

업데이트 함수와 마찬가지로 INDEX 를 기반으로 작동함. 사용자에게 INDEX 를 입력 받아서 그 인덱스를 기준으로 한칸씩 인덱스를 밀어서 덮어쓰움

여기서 INT 부분인 ID 는 그냥 대입으로 처리를 했고 나머지는 전역변수에서 CHAR 형으로 설정했기에 STRCPY 로 덮어쓰웠음. 이후 COUNT 를 감소시킴으로써 스택을 감소시킴..

```

void DELETE(void) { // 제거 함수
    int index; // 업데이트 함수와 마찬가지로 index 를 기반으로 작동
    printf("Please enter the index to be deleted : ");
    scanf("%d", &index);

    if (index >= 0) { // 제거할 인덱스를 입력하면 그 위치를 기반으로 한 칸씩 밀어서 덮어쓰움.
        for (int i = index + 1; i < count; i++) {
            Book_ID[i - 1] = Book_ID[i]; // ID 는 int 형이라서 그냥 대입하면 되는데
            strcpy(Book_Name[i - 1], Book_Name[i]); // 여기는 char 형이라서 C 언어는 string 으로 처리가
            // 안되서 그냥 strcpy 를 사용해서 덮어쓰움.
            strcpy(Book_date_of_issue[i - 1], Book_date_of_issue[i]);
            strcpy(Publisher_Name[i - 1], Publisher_Name[i]);
            strcpy(Publisher_Number[i - 1], Publisher_Number[i]);
            strcpy(Author_Name[i - 1], Author_Name[i]);
            strcpy(Author_eaddress[i - 1], Author_eaddress[i]);
        }
        count--; // 이후 count 를 하나 감소시켜서 스택을 감소시킴.
    } else
        printf("Error -- not found"); // 만약 입력받은 인덱스 음수라면 Error 출력
}

```

## 5) SEARCH 함수

책을 기본적으로 알 때 SEARCH 를 사용하기에 BOOK\_ID 를 입력받아서

INDEX 알려주도록 만들었음. INDEX 번호만 알면 나머지 정보도 알 수 있다고 판단.

FOR 문을 통해 입력받은 ID 가 전역변수에 있는지 확인을 하고 있다면 INDEX 를 출력하고 없다면 ERROR 를 출력하게함.

```
int SEARCH(void) { // 검색 함수
    int BookID;
    printf("Please enter the BOOK ID : "); // BookID 를 입력 받아서 찾기로 함 왜냐면 책 이름만 알때 책
    이름을 입력해서 index 번호를 알아내면 나머지 정보도 얻을 수 있기 때문에.
    scanf("%d", &BookID);

    for (int i = 0; i < count; i++) { // for 문을 돌려서 전역변수에 들어있는지 확인, 없으면 Error 출력, 있으면
    인덱스 번호를 알려줌.
        if (Book_ID[i] == BookID) {
            printf("Existing Book ID | INDEX == %d\n", i);
            return (0); // 여기서 리턴은 큰 의미 없음. 그냥 함수를 종료시키려고 넣은 것.
        }
    }
    printf("Error -- ID not found");
    return (-1); // not found
}
```

## 6) DISPLAY 함수

IF 문을 통해 작가명, 출판사명, 전체를 고를 수 있게 설정했음.

여기서 FINDAUTHRO 와 FINDPUBLISHER 함수를 사용한 이유는 코드가 너무

더러워져서 이기도 했고 기본적으로 전역변수가 CHAR 형이라서 제대로 비교가 안됐기에 포인터를 사용해서 매개변수를 전달했음.

```
void DISPLAY(void) { // 디스플레이 함수.
    char publisher[1000]; // 출판사와 작가 변수를 입력받을 수 있게 변수를 선언함.
    char author[1000];
    int num = 0;
    printf("1. Author\n2. Publisher\n3. All\nEnter the number : ");
    scanf("%d", &num); //여기서 if else 문을 통해서 입력받은 수에 따라서 작가, 출판사, 전체를 골라서 볼 수
    있게 설정함.

    if (num == 1) { // 작가명을 기준으로 확인
        printf("Enter the author name : ");
        scanf("%s", author);
        findauthor((char *)author); // display 함수에서 매개변수를 보낼때 애스터리스크로 보낸 이유는
```

char 형이라 보낼때 이상하게 매개변수 전달이 되서 if 문에 걸리지 않는 오류가 발생했음.

```
} else if (num == 2) { // 출판사명을 기준으로 확인.
```

```
    printf("Enter the Publisher name : ");
```

```
    scanf("%s", publisher);
```

```
    findpublisher((char *)publisher); // display 함수에서 매개변수를 보낼때 애스터리스크로 보낸 이유는
```

char 형이라 보낼때 이상하게 매개변수 전달이 되서 if 문에 걸리지 않는 오류가 발생했음.

```
} else if (num == 3) { // 전체 다 출력
```

```
    printf("Index\tBook ID\tBook Name\tBook's date of issue\tPublisher Name\tPublisher Number\tAuthor  
Name\tAuthor Email Address\n");
```

```
    for (int i = 0; i < count; ++i) { // count 즉 스택이 쌓인 만큼 돌려서 전부 확인
```

```
        printf("%d\t%d\t%s\t%s\t%s\t%s\t%s\n", i, Book_ID[i], Book_Name[i], Book_date_of_issue[i],  
        Publisher_Name[i], Publisher_Number[i], Author_Name[i], Author_eaddress[i]);
```

```
    }
```

```
} else
```

```
    printf("Enter a number from 1 to 3.\n");
```

```
}
```

## 7) CHECK 함수

FOR 문을 통해 매개변수로 전달받은 ID 를 전역변수와 비교해서 있다면 RETURN 1  
없다면 RETURN 0 를 하도록 설정함.

```
int check(int ID) { // 중복조건을 확인해주는 함수
    for (int i = 0; i <
        count; i++) { // 반복문을 count 만큼 돌려 즉 스택의 가장 윗부분 자료가 쌓인 만큼 돌려서 만약
        글로벌 변수안에 중복되는 상황이 발생하면 return 1 을 하고 아니라면 return 0 를 함.
            if (Book_ID[i] == ID)
                return (1); // True
        }
        return (0); // False
    }
```

## 8) FINDAUTHOR, FINDPUBLISHER 함수

각각 작가, 출판사를 기준으로 출력하도록 설정함.

여기서 포인터를 사용해서 전역변수와 매개변수를 비교해서 같다면 출력하도록 설정했음.

```
int findpublisher(char name) { // 출판사명을 기준으로 책정보를 출력해주는 함수
    printf("Index\tBook ID\tBook Name\tBook's date of issue\tPublisher Name\tPublisher Number\tAuthor
    Name\tAuthor Email Address\n");
    for (int i = 0; i < count; i++) {
        if ((char)*Publisher_Name[i] == name) { // 포인터를 사용해서 전역변수와 매개변수를 비교해서 같다면
        출력하도록 설정
            printf("%d\t%d\t%s\t%s\t%s\t%s\t%s\t%s\n", i, Book_ID[i], Book_Name[i], Book_date_of_issue[i],
            Publisher_Name[i], Publisher_Number[i], Author_Name[i], Author_eaddress[i]);
        }
    }
    return (0); // not found
}
```