

1. struct를 선언하고 사용하는 방법:

```
struct 변수명{  
    변수1;  
    변수2;  
};
```

와 같이 변수를 선언할 수 있고, typedef를 사용하면 struct를 사용할 때, struct book으로 사용할 필요없이 book만을 써서 struct를 쓸 수 있다.

예를 들어

```
typedef struct{  
    char name[30];  
    char semi_ISBN[5];  
    int page;  
}book;
```

라는 struct book이 있다면, 우선 다음과 같이 book에 대한 메모리를 할당을 해주고,

```
book *shelf =(book *) malloc(10*sizeof(book)); //메모리 할당
```

. -> 연산자를 이용해서 struct를 사용할 수 있다.

```
EX) *shelf.name    (*(shelf+2)).name
```

```
EX) strcpy((shelf+2)->name, "I love you");
```

```
EX) printf("%s", (shelf+4)->semi_ISBN);
```

2. struct를 사용하는 목적:

우선 struct의 정의에 대해 알아야한다. struct란 하나이상의 변수를 묶어서 새로운 자료형을 정의하는 도구이다. 그래서 struct를 사용하는 이유는 주로 여러가지 관련된 변수들을 하나로 묶어서 개발자로 하여금 보기 쉽게해준다. 또한 관련된 변수들을 추가해야 할 때, 쉽게 추가할 수 있고, 정리하기도 편하다.

3. 프로그램의 설계과정:

우선 메뉴를 만들어야했다. 메뉴는 5를 입력받기 전까지는 계속 생성되어야했다. 입력값에 대한 예외처리가 필요했다.

책정보를 보여주는 것은 우선 책이 몇권이 있는지를 파악을 하고 정보를 권수에맞게 보여주는 것이 필요할 것이라고 생각했다.

책정보를 추가할 때에 우선 메모리 값을 유동적으로 활용할줄 알아야했고, name, semi_ISBN, page에 대한 정보를 받을 때 그 값의 범위들과 예외처리를 생각했다.

책정보를 삭제할 때 우선 입력값에 대한 예외처리를 먼저 생각했고, 하나의 정보를 삭제하면 그

비어있는 자리를 뒤에 있는 정보들을 한개씩 끌어와 붙이고 메모리를 하나 해제해줘야겠다고 생각했었다.

insensitive한 알파벳순 정렬은 버블솔트를 이용하기로 했다. insensitive하게 만들려면 각각의 문자열의 문자값들이 대문자인지, 소문자인지, 영문이 아닌지를 판단하고 변환하는 방법을 고안해야 했다.

프로그램을 종료시키려면 계속 생성되는 메뉴를 그만 생성되게 만들어야했다.

4. 프로그램의 작동원리:

i. struct를 선언을 하고, int index를 선언한다. 구조체 shelf를 선언을 해주고, index만큼 malloc해 준다. int level=-1로 선언, 초기화했다 shelf 구조체에 {"Let's do it", "4456", 36}를 일단 넣어주었다. menu를 설계했다. 일단 int범위 예외처리를 생각했다. INT_MAX=+2147483647인데 어떤 정수형이 되었던간에 범위를 넘어가면 음수로 값이 순환되어 버림으로 정수형으로 선언을 할 수 없었다. 어쩔 수 없이 char 배열을 생성해서 Input을 받을 수 밖에 없었다. INT_MAX이 10자리라서 우선 10자리 char 배열을 받는 함수를 만드는데, 예외처리를 하기위해 getchar()와 for문으로 받았다. 만일 10자리가 될때 까지 'Wn'이 안나타나면, while(getchar()!='Wn');으로 버퍼값을 삭제했다. 그런 함수를 구현한 다음에 각각의 char이 숫자인지를 판단하고, 나중에 long값으로 옮겨서 하나의 정수로 만들고, 1~5사이에 있는지를 판단했다.

Display를 하는 방식은 index의 값을 보여주고, for문을 이용해서 각각의 book에 대한 정보들 (name, semi_ISBN , page)을 보여준다.

book을 추가하는 것은 다음과 같다.

1. index++;을 해주고 shelf를 realloc해주어 book의 수를 더 늘린다.
2. name: 아까만든 getchar()와 for문으로 문자열 받는 함수로 30자를 받아주고
3. semi_ISBN: 5자리 char 배열을 만들어주고 4개의 숫자와 'W0'이 잘들어갔는지 확인해준다.
4. page: 메뉴를 만들때 사용한 함수를 동일하게 사용해서 1~INT_MAX사이의 정수값을 받아준다.

book을 제거하는 방법은

1. index--;를 해준다.
2. book fix의 name에 getchar()와 for문으로 문자열 받는 함수로 30자를 받아주고
3. for문으로 index를 하나하나 확인하면서 strcmp로 fix와 shelf[i]의 값을 비교한다.
4. 값이 같으면 그 자리에 뒤에 나오는 *(shelf+i+1)을 *(shelf +i)에 넣어버린다.
5. realloc으로 메모리 값을 줄여준다.

book을 버블솔트를 이용해서 알파벳순으로 insensitive하게 정렬시키려면

1. 일단 for문을 두개 선언하고 안쪽에 들어간 for문이 끝날 때마다 안쪽의 for문의 조건절의 범위를 0에 가깝게 한개씩 줄여준다. (안쪽의 for문을 한번 돌리면 맨뒤의 값부터 차례대로 sort되기 때문이다.)
2. 두개의 for문 안에 일일이 name[i]를 확인하면서 bucket1->name에 복사한다. 그 과정에서 대문자가 있으면 소문자로 변환한다.
3. shelf->name[i]를 bucket1->name에 위와 같은 방식으로 복사하고, shelf->name[i+1]를 bucket2->name에 위와 같은 방식으로 복사한다.
4. 만일 strcmp(bucket1->name, bucket2->name)==1이면, shelf->name[i]이 shelf->name[i+1]보다 알파벳 순으로 뒤에 있다는 이야기이니, book tmp를 이용하여 구조체를 통째로 swap해준다.

exist는 처음에 menu만들때 level==1이면 break;한다라는 코드를 짜놓으면 된다.

5.어려웠던점:

어려웠던점은 바로 예외처리였다. 예외처리 자체가 어려운것이 아니라, 어디까지 무엇을 예외처리 해야하는 지, 즉 예외처리에 대한 기준점이 너무나도 불분명한 것이 힘들었다. 문자열의 갯수가 30인데 40을 입력했을 경우에 대한 예외처리는 충분히 생각할만하다. 그러나 문자열의 크기가 1000000개이고, 1000001만큼 문자를 받는 경우도 예외처리를 해야하나? int정수를 하나 받아도 int값의 범위를 고려하고, int 범위를 넘어가면 다른 이상한 값이 되어버린다는 것도 고려하고, 이런식으로 계속 예외처리들을 고려하다보면 사실 끝이 없다. 혹시나 콘솔창에 한자나 아랍어를 입력하는 경우조차 예외처리를 해야하는가? 과제를 하는 학생의 입장에서 어디서부터 어디까지 예외처리를 하라는 명백한 기준점, 채점기준이 있었으면 좋겠다는 바램이다. 나는 최선을 다해서 내가 생각할 수 있는 모든 예외들을 다 처리했으나, 그게 전부가 아니었고 이런 부분들도 고려를 했어야한다고 나중에 발표되면, 내가 그런 부분들을 예외처리할 능력이 있음에도 불구하고, 일단 점수부터 감점시키고 내 실력을 제대로 발휘하지 못하게 한다면, 나는 진심으로 억울할 것 같다.