**SOURCE CODE**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct Book {

char author[100];

char title[100];

int year\_published;

} Book;

int compare\_strings(const char \*s1, const char \*s2) {

return strcmp(s1, s2);

}

Book \*\*sort\_item\_list\_by\_author(Book \*\*list, int count) {

Book \*\*sorted\_list = (Book \*\*) malloc(count \* sizeof(Book \*));

for (int i = 0; i < count; i++) {

sorted\_list[i] = list[i];

}

for (int i = 0; i < count - 1; i++) {

for (int j = i + 1; j < count; j++) {

if (compare\_strings(sorted\_list[i]->author, sorted\_list[j]->author) > 0) {

Book \*temp = sorted\_list[i];

sorted\_list[i] = sorted\_list[j];

sorted\_list[j] = temp;

}

}

}

return sorted\_list;

}

void sort\_items\_by\_published\_year(Book \*\*list, int count) {

for (int i = 0; i < count - 1; i++) {

for (int j = i + 1; j < count; j++) {

if (list[i]->year\_published > list[j]->year\_published) {

Book \*temp = list[i];

list[i] = list[j];

list[j] = temp;

}

}

}

}

void add\_new\_items(Book \*\*existing\_list, int existing\_count, Book \*\*new\_list, int

new\_count) {

Book \*\*merged\_list = (Book \*\*) malloc((existing\_count + new\_count) \*

sizeof(Book \*));

int i = 0, j = 0, k = 0;

while (i < existing\_count && j < new\_count) {

if (compare\_strings(existing\_list[i]->author, new\_list[j]->author) < 0) {

merged\_list[k++] = existing\_list[i++];

} else {

merged\_list[k++] = new\_list[j++];

}

}

while (i < existing\_count) {

merged\_list[k++] = existing\_list[i++];

}

while (j < new\_count) {

merged\_list[k++] = new\_list[j++];

}

merged\_list = sort\_item\_list\_by\_author(merged\_list, existing\_count +

new\_count);

sort\_items\_by\_published\_year(merged\_list, existing\_count + new\_count);

printf("List of books in the alphabetical order of author:\n");

for (i = 0; i < existing\_count + new\_count; i++) {

printf("%s - %s (%d)\n", merged\_list[i]->author, merged\_list[i]->title,

merged\_list[i]->year\_published);

}

free(merged\_list);

}

int main() {

int existing\_count;

printf("Enter the number of existing books: ");

scanf("%d", &existing\_count);

Book \*\*existing\_list = (Book \*\*) malloc(existing\_count \* sizeof(Book \*));

printf("Enter the details of existing books (author name, book title, year

published):\n");

for (int i = 0; i < existing\_count; i++) {

existing\_list[i] = (Book \*) malloc(sizeof(Book));

scanf("%s%s%d", existing\_list[i]->author, existing\_list[i]->title, &existing\_list[i]-

>year\_published);

}

existing\_list = sort\_item\_list\_by\_author(existing\_list, existing\_count);

sort\_items\_by\_published\_year(existing\_list, existing\_count);

int new\_count;

printf("Enter the number of new books: ");

scanf("%d", &new\_count);

Book \*\*new\_list = (Book \*\*) malloc(new\_count \* sizeof(Book \*));

printf("Enter the details of new books (author name, book title, year published):\n");

for (int i = 0; i < new\_count; i++) {

new\_list[i] = (Book \*) malloc(sizeof(Book));

scanf("%s%s%d", new\_list[i]->author, new\_list[i]->title, &new\_list[i]-

>year\_published);

}

new\_list = sort\_item\_list\_by\_author(new\_list, new\_count);

sort\_items\_by\_published\_year(new\_list, new\_count);

add\_new\_items(existing\_list, existing\_count, new\_list, new\_count);

for (int i = 0; i < existing\_count; i++) {

free(existing\_list[i]);

}

free(existing\_list);

for (int i = 0; i < new\_count; i++) {

free(new\_list[i]);

}

free(new\_list);

return 0;

}