HackerRank | Prepare Certify Compete Apply Search

All Contests > CSE-g-week2 > Book Inventory Manager

Book Inventory Manager

Problem Submissions Leaderboard Discussions

Create a simple inventory management system for a small bookstore:

1) Define a Book structure with the following fields:

Title: a string with a maximum of 100 characters
Author: a string with a maximum of 50 characters
Price: a float

2) Implement an AddBook function that dynamically allocates memory for an array of Book structures. This function should allow the user to add a new book to the inventory. If the inventory reaches its capacity, the function should double the memory allocation to accommodate more books.

3) Create a DisplayInventory function that takes the array of books and the number of books in the inventory as parameters. This function should display the details of all books in the inventory in a single line, space-separated in the following order: Title, Author, Price. The price should be formatted to two decimal places. If the inventory is empty, the function should return -1.

Ensure all dynamically allocated memory is properly freed.

Input Format

line1: The first line contains an integer n, the number of operations to be performed.
line2: For each operation, the input will be either:
1 (to add a new book), followed by the details of the book: title, author and price each detail input is taken in a new line.
2 (to display the inventory).

Constraints

0<=Title[i]<=100
0<=Author[i]<=50

Output Format

For each DisplayInventory operation (2), print the details of all books in the inventory. Each book's details should be printed on a single line, with the fields title, author and price space-separated.

Sample Input 0

```
5
1
The_C_Programming_Language
Brian_W._Kernighan_and_Dennis_M._Ritchie
45.99
1
Clean_Code
Robert_C._Martin
37.95
2
1
The_Pragmatic_Programmer
Andrew_Hunt_and_David_Thomas
42.50
2
```

Sample Output 0

```
The_C_Programming_Language Brian_W._Kernighan_and_Dennis_M._Ritchie 45.99
Clean_Code Robert_C._Martin 37.95
The_C_Programming_Language Brian_W._Kernighan_and_Dennis_M._Ritchie 45.99
Clean_Code Robert_C._Martin 37.95
The_Pragmatic_Programmer Andrew_Hunt_and_David_Thomas 42.50
```

Contest ends in a day

Submissions: 11

Max Score: 8

Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆

More

C

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define INITIAL_CAPACITY 2

typedef struct book {
    char title[100];
    char author[50];
    float price;
} Book;

void AddBook(Book **inventory, int *count, int *capacity) {
    if (*count >= *capacity) {
        *capacity *= 2;
        *inventory = realloc(*inventory, (*capacity) * sizeof(Book));
        if (*inventory == NULL) {
            exit(1);
        }
    }

    scanf(" %[^\n]%*c", (*inventory)[*count].title);
    scanf(" %[^\n]%*c", (*inventory)[*count].author);
    scanf("%f", &(*inventory)[*count].price);

    (*count)++;
}

void DisplayInventory(Book *inventory, int count) {
    if(count==0){printf("%d\n",-1);}
    else{
        for (int i = 0; i < count; i++) {
        printf("%s %s %.2f\n", inventory[i].title, inventory[i].author, inventory[i].price);
    }
    }
}

int main() {
    int n;
    scanf("%d", &n);

    int choice;
    int count = 0;
    int capacity = INITIAL_CAPACITY;

    Book *arr = (Book*)malloc(capacity * sizeof(Book));
    if (arr == NULL) {
        return 1;
    }

    for (int i = 0; i < n; i++) {
        scanf("%d", &choice);

        if (choice == 1) {
            AddBook(&arr, &count, &capacity);
        } else if (choice == 2) {
            DisplayInventory(arr, count);
        } else {
            break;
        }
    }

    free(arr);
    return 0;
}
```

Line: 1 Col: 1

Upload Code as File

□ Test against custom input

Run Code

Submit Code

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |