```
// Experiment 3: Implement Stack using Array: Bookstack Management System
// WAP to develop a simple management system for a library that helps track the books returned by patrons.
// Each book returned should be pushed onto a stack, and you need to implement this functionality using both an array.
// Menu for the Program:
// Welcome to the Bookstack Management System!
// 1. Add a book
// 2. Remove the most recent book
// 3. View the most recent book
// 4. Display all books
// 5. Exit
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

char **stack;  // Array of pointers to store book names
int top = -1;
int size = 0;

// Function to create a stack
void createStack(int isize) {
    size = isize;
    stack = (char **)malloc(size * sizeof(char *));  // Array of char pointers for book names
    if (!stack) {
        exit(0);
    }
}

// Check if the stack is full
int isfull() {
    return (top == size - 1);
}

// Check if the stack is empty
int isempty() {
    return (top == -1);
}

void resizeStack(){
        size = size*2;
        stack = (char **)realloc(stack,size*sizeof(char*));
        if(!stack){
                exit(0);
        }
}
```

```
// Function to push a book into the stack
void push(char val[]) {
    if (isfull()) {
        resizeStack();
    } else {
        top = top + 1;
        stack[top] = (char *)malloc((strlen(val) + 1) * sizeof(char));  // Allocate memory for the
book name
        strcpy(stack[top], val);  // Copy the book name to the stack
        printf("\n '%s' is Successfully added \n", val);
    }
}

// Function to pop a book from the stack
void pop() {
    if (isempty()) {
        printf("\n Book not Found ! \n");
    } else {
        printf("\n '%s' is Successfully Provided! \n", stack[top]);
        free(stack[top]);  // Free the memory of the book name
        top = top - 1;
    }
}

// Function to show  the top book
void peek() {
    if (!isempty()) {

        printf("%s\n", stack[top]);
    } else {

        printf("No books available!\n");
    }
}

// to show all books in the stack
void show() {
    if (!isempty()) {
        printf("Books are:- \n");
        for (int i = 0; i <= top; i++) {
            printf("%s\n", stack[i]);
        }
    } else {
        printf("No books in the shelf!\n");
    }
}
```

```
int main() {
    createStack(4);

    printf("**************************************************\n");
    printf("1. Add a returned Book\n");
    printf("2. Remove a returned Book\n");
    printf("3. View the top returned Book\n");
    printf("4. View all the books\n");
    printf("5. Exit\n");

    char name[200];
    while (1) {
        int choice;
        printf("\nEnter your choice: ");
        scanf("%d", &choice);
        getchar();

        switch (choice) {
            case 1:
                printf("Enter the book name: ");
                fgets(name, 200, stdin);
                name[strcspn(name, "\n")] = 0;  // Remove  newline from fgets
                push(name);
                break;

            case 2:
                pop();
                break;

            case 3:
                printf("The recent book is: \n");
                peek();
                break;

            case 4:
                show();
                break;

            case 5:
                printf("Thank you for your visit!\n");
                exit(0);
        }

        printf("**************************************************\n");
    }

    return 0;
}
```