

Rajalakshmi Engineering College

Name: kamali rj
Email: 240701225@rajalakshmi.edu.in
Roll no: 240701225
Phone: 9344843996
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 2.5

Section 1 : Coding

1. Problem Statement

Sanjeev is in charge of managing a library's book storage, and he wants to create a program that simplifies this task. His goal is to implement a program that simulates a stack using an array.

Help him in writing a program that provides the following functionality:

Add Book ID to the Stack (Push): You can add a book ID to the top of the book stack. Remove Book ID from the Stack (Pop): You can remove the top book ID from the stack and display its details. If the stack is empty, you cannot remove any more book IDs. Display Books ID in the Stack (Display): You can view the books ID currently on the stack. Exit the Library: You can choose to exit the program.

Input Format

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Push the book onto the stack. If the choice is 1, the following input is a space-separated integer, representing the ID of the book to be pushed onto the stack.

Choice 2: Pop the book ID from the stack.

Choice 3: Display the book ID in the stack.

Choice 4: Exit the program.

Output Format

The output displays messages according to the choice and the status of the stack:

1. If the choice is 1, push the given book ID to the stack and display the corresponding message.
2. If the choice is 2, pop the book ID from the stack and display the corresponding message.
3. If the choice is 2, and if the stack is empty without any book ID, print "Stack Underflow"
4. If the choice is 3, print the book IDs in the stack.
5. If the choice is 3, and there are book IDs in the stack, print "Stack is empty"
6. If the choice is 4, exit the program and display the corresponding message.
7. If any other choice is entered, print "Invalid choice"

Refer to the sample output for the exact text and format.

Sample Test Case

Input: 1 19

1 28

2

3

2

4

Output: Book ID 19 is pushed onto the stack

Book ID 28 is pushed onto the stack

Book ID 28 is popped from the stack
Book ID in the stack: 19
Book ID 19 is popped from the stack
Exiting the program

Answer

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_CAPACITY 100
typedef struct {
    int stack[MAX_CAPACITY];
    int top;
} BookStack;
void initializeStack(BookStack *s) {
    s->top = -1;
}
int isFull(BookStack *s) {
    return s->top == MAX_CAPACITY - 1;
}
int isEmpty(BookStack *s) {
    return s->top == -1;
}
void push(BookStack *s, int bookId) {
    if (isFull(s)) {
        printf("Stack overflow\n");
    } else {
        s->stack[++s->top] = bookId;
        printf("Book ID %d is pushed onto the stack\n", bookId);
    }
}
int pop(BookStack *s) {
    if (isEmpty(s)) {
        printf("Stack underflow\n");
        return -1; // Indicate an error or empty stack
    } else {
        int bookId = s->stack[s->top--];
        printf("Book ID %d is popped from the stack\n", bookId);
        return bookId;
    }
}
void display(BookStack *s) {
    if (isEmpty(s)) {
```

```

        printf("Stack is empty\n");
    } else {
        printf("Book ID in the stack: ");
        for (int i = 0; i <= s->top; i++) {
            printf("%d", s->stack[i]);
            if (i < s->top) {
                printf(", ");
            }
        }
        printf("\n");
    }
}

int main() {
    BookStack bookStack;
    initializeStack(&bookStack);
    int choice, bookId;

    while (1) {
        if (scanf("%d", &choice) != 1) {
            printf("Invalid choice\n");
            while (getchar() != '\n');
            continue;
        }

        switch (choice) {
            case 1:
                if (scanf("%d", &bookId) == 1) {
                    push(&bookStack, bookId);
                } else {
                    printf("Invalid input\n");
                    while (getchar() != '\n');
                }
                break;
            case 2:
                pop(&bookStack);
                break;
            case 3:
                display(&bookStack);
                break;
            case 4:
                printf("Exiting the program\n");
                return 0;
        }
    }
}

```

```
        default:
            printf("Invalid choice\n");
        }
    }

    return 0;
}
```

Status : Partially correct

Marks : 2.5/10