Rajalakshmi Engineering College

Name: LIRESH NV
Email: 241501100@rajalakshmi.edu.in
Roll no: 241501100
Phone: 9840466142
Branch: REC
Department: I AI & ML FA
Batch: 2028
Degree: B.E - AI & ML

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 : 10

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

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>

#define MAX_SIZE 100

typedef struct Stack {
   int books[MAX_SIZE];
   int top;
} Stack;

// Function to initialize the stack
void initStack(Stack* s) {
   s->top = -1;
}

// Function to check if the stack is full
int isFull(Stack* s) {
   return s->top == MAX_SIZE - 1;
}

// Function to check if the stack is empty
int isEmpty(Stack* s) {
   return s->top == -1;
}

// Function to push a book ID onto the stack
void push(Stack* s, int bookID) {
   if (isFull(s)) {
      printf("Stack is full. Cannot push book ID %d\n", bookID);
      return;
   }
   s->books[++(s->top)] = bookID;
   printf("Book ID %d is pushed onto the stack\n", bookID);
}
```

```
// Function to pop a book ID from the stack
void pop(Stack* s) {
    if (isEmpty(s)) {
        printf("Stack Underflow\n");
        return;
    }
    int poppedBookID = s->books[(s->top)--];
    printf("Book ID %d is popped from the stack\n", poppedBookID);
}

// Function to display the book IDs in the stack
void display(Stack* s) {
    if (isEmpty(s)) {
        printf("Stack is empty\n");
        return;
    }
    printf("Book ID in the stack: ");
    for (int i = s->top; i >= 0; i--) {
        printf("%d ", s->books[i]);
    }
    printf("\n");
}

int main() {
    Stack stack;
    initStack(&stack);
    int choice, bookID;

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

        switch (choice) {
            case 1:
                if (scanf("%d", &bookID) != 1) {
                    printf("Invalid choice\n");
                    // Clear invalid input
                    while (getchar() != '\n');
                    continue;
                }
                push(&stack, bookID);
                break;
            case 2:
                pop(&stack);
                break;
            case 3:
                display(&stack);
                break;
            case 4:
                printf("Exiting the program\n");
                return 0;
            default:
                printf("Invalid choice\n");
                break;
        }
    }
    return 0;
}
```

Status : Correct ***Marks : 10/10***