3) Stack is a simple linear data structure used for storing data which follows the principle of Last In First Out. Assume that you are given the details of BOOK with members ISBN, Title, Author and Price. Design an interactive C program to construct a stack data structure to store N BOOK items and write C functions to perform the following operations on it: a. PUSH-To add a new BOOK to the stack b. POP- To remove a BOOK from the stack Also demonstrate Overflow and Underflow conditions on Stack and display the status of Stack.

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define SIZE 5 // Maximum size of the stack


// Define structure for a book

struct BOOK {

   int ISBN;            // ISBN number of the book

   char title[50];      // Title of the book

   char author[50];     // Author of the book

   float price;         // Price of the book

};


// Define structure for a stack of books

struct stack {

   struct BOOK b[SIZE];  // Array to store books

   int top;             // Top index of the stack

};


// Function prototypes

void push(struct stack *ps, struct BOOK b1);

struct BOOK pop(struct stack *ps);

void display(struct stack *ps);


void main() {
```

```
struct stack s;        // Declare a stack of books
struct BOOK b1, r1;    // Declare variables for book operations
int choice;
s.top = -1;            // Initialize top of stack

do {
   // Display menu
   printf("\n1: PUSH\t 2: POP\t 3: DISPLAY\t 4: QUIT");
   printf("\nEnter your choice: ");
   scanf("%d", &choice);

   switch (choice) {
      case 1:
         // Prompt user to enter book details
         printf("Enter the ISBN, title, author, and price of the book to push:\n");
         scanf("%d", &b1.ISBN);
         getchar();  // To clear the newline character left by scanf
         printf("Enter Title: ");
         fgets(b1.title, sizeof(b1.title), stdin);
         b1.title[strcspn(b1.title, "\n")] = '\0'; // Remove newline character

         printf("Enter Author: ");
         fgets(b1.author, sizeof(b1.author), stdin);
         b1.author[strcspn(b1.author, "\n")] = '\0'; // Remove newline character

         printf("Enter Price: ");
         scanf("%f", &b1.price);

         push(&s, b1);  // Push the book onto the stack
         break;
```

```
        case 2:
            r1 = pop(&s);  // Pop the top book from the stack
            printf("The details of BOOK popped are:\n");
            printf("ISBN = %d, Title = %s, Author = %s, Price = %f\n",
                   r1.ISBN, r1.title, r1.author, r1.price);
            break;

        case 3:
            display(&s);  // Display the books in the stack
            break;

        case 4:
            printf("\nQuitting operation stack\n");
            break;

        default:
            printf("No such option\n");
            break;
        }
    } while (choice != 4);  // Repeat until user chooses to quit
}

// Function to push a book onto the stack
void push(struct stack *ps, struct BOOK b1) {
    if (ps->top == SIZE - 1) {  // Check if the stack is full
        printf("Stack Overflow\n");
    } else {
        ++(ps->top);              // Increment top index
        ps->b[ps->top] = b1;     // Add the book to the stack
        printf("Book pushed onto the stack.\n");
    }
}

// Function to pop a book from the stack
struct BOOK pop(struct stack *ps) {
    struct BOOK r;
    if (ps->top == -1) {  // Check if the stack is empty
        printf("\nStack Underflow\n");
        exit(1);  // Exit the program if the stack is empty
    } else {
        r = ps->b[ps->top];  // Get the top book
        (ps->top)--;         // Decrement the top index
        return r;            // Return the popped book
    }
}

// Function to display the contents of the stack
void display(struct stack *ps) {
    int i;
    if (ps->top == -1) {  // Check if the stack is empty
        printf("\nStack is empty\n");
    } else {
        printf("Stack contents are:\n");
        for (i = ps->top; i >= 0; i--) {  // Iterate over the stack from top to bottom
            printf("ISBN = %d, Title = %s, Author = %s, Price = %.2f\n",
                   ps->b[i].ISBN, ps->b[i].title, ps->b[i].author, ps->b[i].price);
        }
    }
}
```