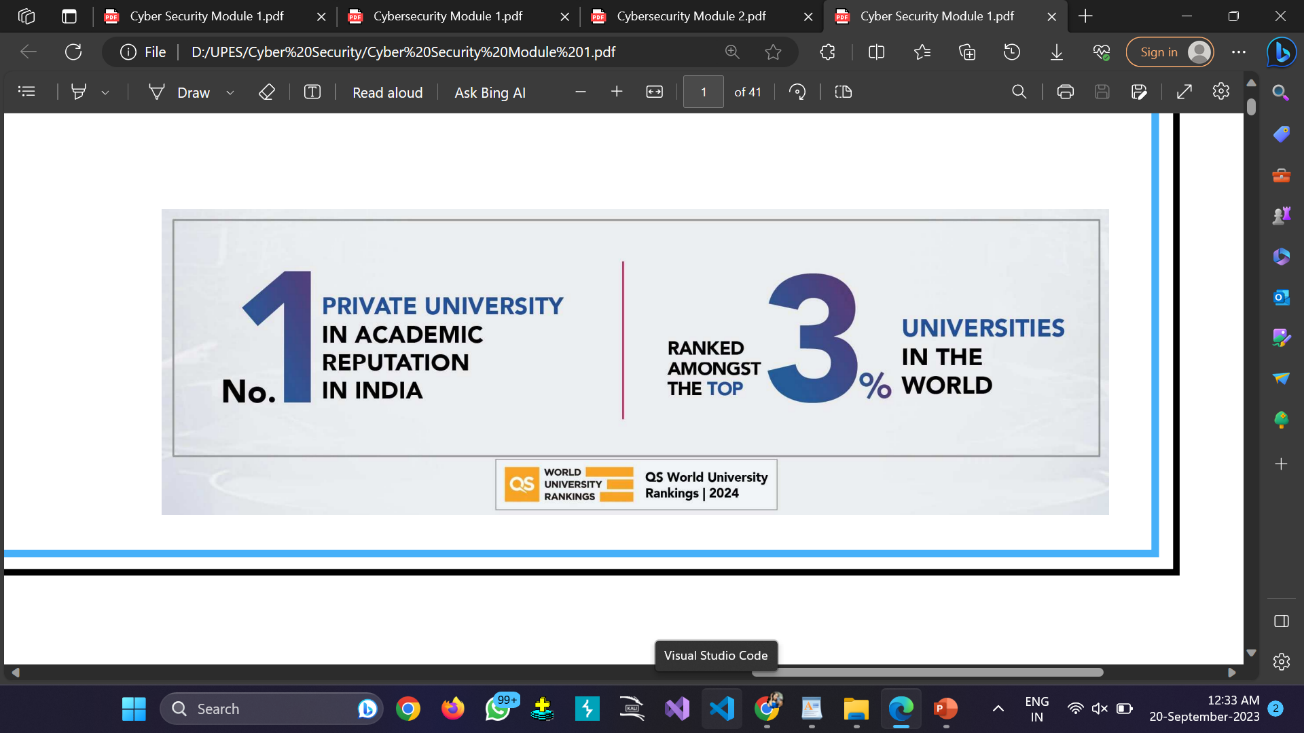
A picture containing text, clipart

Description automatically generated

1

**Lab Experiment: 04**

**Student Detail:**

**• Name:** Prashant Joshi

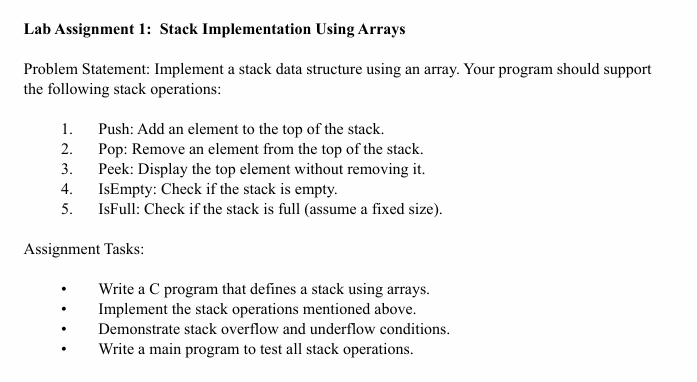
**• Student ID:** 590010879

**• Branch:** MCA

**• Batch:** B1

**• Instructor:** Dr. Sourbh Kumar

**Lab Assignment 1: Stack Implementation Using Arrays**



Solution:

#include <stdio.h>

#define MAX 5 *// Define the maximum size of the stack*

int stack[MAX];

int top = -1; *// Initial stack is empty*

*// Function to add an element to the stack*

void push(int value) {

    if (top == MAX - 1) {

        printf("Stack Overflow! Cannot push %d\n", value);

    } else {

        stack[++top] = value;

        printf("%d pushed to the stack\n", value);

    }

}

*// Function to remove the top element from the stack*

void pop() {

    if (top == -1) {

        printf("Stack Underflow! Cannot pop\n");

    } else {

        printf("%d popped from the stack\n", stack[top--]);

    }

}

*// Function to display the top element of the stack*

void peek() {

    if (top == -1) {

        printf("Stack is empty\n");

    } else {

        printf("Top element is %d\n", stack[top]);

    }

}

*// Function to check if the stack is empty*

int isEmpty() {

    return top == -1;

}

*// Function to check if the stack is full*

int isFull() {

    return top == MAX - 1;

}

*// Main function to test the stack operations*

int main() {

    int choice, value;

    do {

        printf("\nStack Operations:\n");

        printf("1. Push\n");

        printf("2. Pop\n");

        printf("3. Peek\n");

        printf("4. Check if Empty\n");

        printf("5. Check if Full\n");

        printf("6. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice) {

            case 1:

                printf("Enter value to push: ");

                scanf("%d", &value);

                push(value);

                break;

            case 2:

                pop();

                break;

            case 3:

                peek();

                break;

            case 4:

                if (isEmpty()) {

                    printf("Stack is empty\n");

                } else {

                    printf("Stack is not empty\n");

                }

                break;

            case 5:

                if (isFull()) {

                    printf("Stack is full\n");

                } else {

                    printf("Stack is not full\n");

                }

                break;

            case 6:

                printf("Exiting program\n");

                break;

            default:

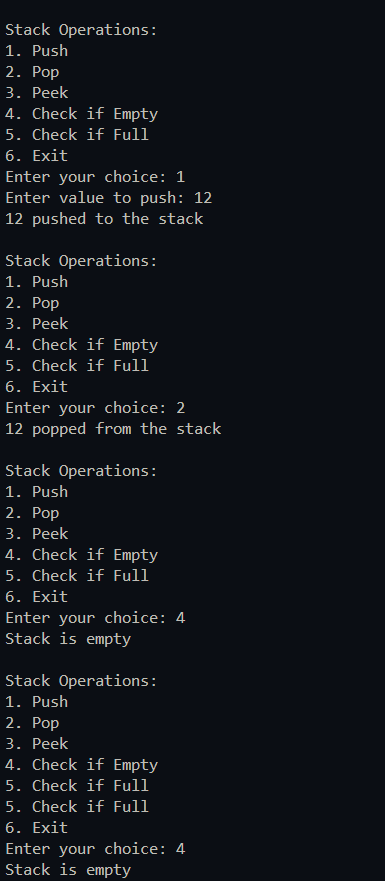
                printf("Invalid choice! Please try again.\n");

        }

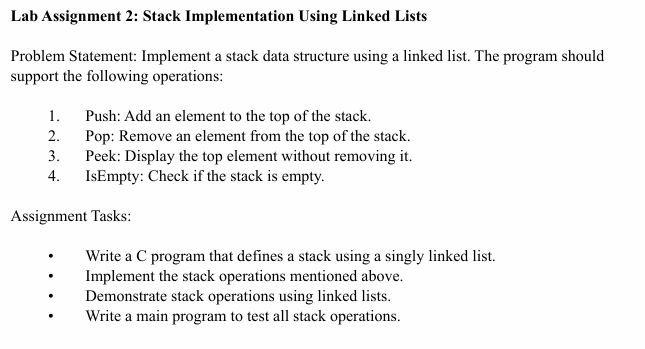
    } while (choice != 6);

    return 0;

}

Output:

Lab Assignment 2:



Solution:

#include <stdio.h>

#include <stdlib.h>

// Define a node structure for the stack

struct Node {

int data;

struct Node\* next;

};

// Top of the stack

struct Node\* top = NULL;

// Function to check if the stack is empty

int isEmpty() {

return top == NULL;

}

// Function to add an element to the top of the stack

void push(int value) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

if (!newNode) {

printf("Heap Overflow! Cannot push %d\n", value);

return;

}

newNode->data = value;

newNode->next = top;

top = newNode;

printf("%d pushed to the stack\n", value);

}

// Function to remove the top element from the stack

void pop() {

if (isEmpty()) {

printf("Stack Underflow! Cannot pop\n");

return;

}

struct Node\* temp = top;

printf("%d popped from the stack\n", top->data);

top = top->next;

free(temp);

}

// Function to display the top element of the stack

void peek() {

if (isEmpty()) {

printf("Stack is empty\n");

} else {

printf("Top element is %d\n", top->data);

}

}

// Main function to test the stack operations

int main() {

int choice, value;

do {

printf("\nStack Operations:\n");

printf("1. Push\n");

printf("2. Pop\n");

printf("3. Peek\n");

printf("4. Check if Empty\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter value to push: ");

scanf("%d", &value);

push(value);

break;

case 2:

pop();

break;

case 3:

peek();

break;

case 4:

if (isEmpty()) {

printf("Stack is empty\n");

} else {

printf("Stack is not empty\n");

}

break;

case 5:

printf("Exiting program\n");

break;

default:

printf("Invalid choice! Please try again.\n");

}

} while (choice != 5);

return 0;

}

Output:

