***NAME : JANHAVI GATTANI***

***BATCH :2***

***ROLL NO:37***

***PRN:12311291***

***DS LAB ASSIGNMENT 5***

**QUESTION : Write a program to implement Stack and Queue basic operations**

#include <stdio.h>

#define SIZE 5

typedef struct {

int top;

int arr[SIZE];

} Stack;

void initStack(Stack \*s) {

s->top = -1;

}

void push(Stack \*s, int value) {

if (s->top == SIZE - 1) {

printf("Stack Overflow\n");

} else {

s->top++;

s->arr[s->top] = value;

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

}

}

void pop(Stack \*s) {

if (s->top == -1) {

printf("Stack Underflow\n");

} else {

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

s->top--;

}

}

void displayStack(Stack \*s) {

if (s->top == -1) {

printf("Stack is empty\n");

} else {

printf("Stack elements: ");

for (int i = s->top; i >= 0; i--) {

printf("%d ", s->arr[i]);

}

printf("\n");

}

}

typedef struct {

int front, rear;

int arr[SIZE];

} Queue;

void initQueue(Queue \*q) {

q->front = -1;

q->rear = -1;

}

void enqueue(Queue \*q, int value) {

if (q->rear == SIZE - 1) {

printf("Queue Overflow\n");

} else {

if (q->front == -1) q->front = 0;

q->rear++;

q->arr[q->rear] = value;

printf("%d enqueued to queue\n", value);

}

}

void dequeue(Queue \*q) {

if (q->front == -1 || q->front > q->rear) {

printf("Queue Underflow\n");

} else {

printf("%d dequeued from queue\n", q->arr[q->front]);

q->front++;

if (q->front > q->rear) q->front = q->rear = -1; // Reset queue

}

}

void displayQueue(Queue \*q) {

if (q->front == -1) {

printf("Queue is empty\n");

} else {

printf("Queue elements: ");

for (int i = q->front; i <= q->rear; i++) {

printf("%d ", q->arr[i]);

}

printf("\n");

}

}

int main() {

Stack stack;

Queue queue;

initStack(&stack);

initQueue(&queue);

printf("Stack Operations:\n");

push(&stack, 10);

push(&stack, 20);

displayStack(&stack);

pop(&stack);

displayStack(&stack);

pop(&stack);

pop(&stack);

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

enqueue(&queue, 100);

enqueue(&queue, 200);

displayQueue(&queue);

dequeue(&queue);

displayQueue(&queue);

enqueue(&queue, 300);

enqueue(&queue, 900);

enqueue(&queue, 400);

enqueue(&queue, 500);

return 0;

}

***OUTPUT:***

