reg no

COM/B/01-00136/2022

COM/B/01-00115/2022

COM/B/01-00110/2022

COM/B/01-00116/2022

#include <stdio.h>

#include <stdlib.h>

// Stack implementation using array

#define MAX\_SIZE 100

typedef struct {

int items[MAX\_SIZE];

int top;

} StackArray;

void initStackArray(StackArray \*stack) {

stack->top = -1;

}

int isEmptyStackArray(StackArray \*stack) {

return (stack->top == -1);

}

int isFullStackArray(StackArray \*stack) {

return (stack->top == MAX\_SIZE - 1);

}

void pushStackArray(StackArray \*stack, int value) {

if (!isFullStackArray(stack)) {

stack->items[++stack->top] = value;

} else {

printf("Stack overflow!\n");

}

}

int popStackArray(StackArray \*stack) {

if (!isEmptyStackArray(stack)) {

return stack->items[stack->top--];

} else {

printf("Stack underflow!\n");

return -1;

}

}

// Queue implementation using linked list

typedef struct Node {

int data;

struct Node\* next;

} Node;

typedef struct {

Node \*front, \*rear;

} QueueLL;

Node\* createNode(int data) {

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

newNode->data = data;

newNode->next = NULL;

return newNode;

}

void initQueueLL(QueueLL\* queue) {

queue->front = queue->rear = NULL;

}

int isEmptyQueueLL(QueueLL\* queue) {

return (queue->front == NULL);

}

void enqueueLL(QueueLL\* queue, int value) {

Node\* newNode = createNode(value);

if (isEmptyQueueLL(queue)) {

queue->front = queue->rear = newNode;

} else {

queue->rear->next = newNode;

queue->rear = newNode;

}

}

int dequeueLL(QueueLL\* queue) {

if (!isEmptyQueueLL(queue)) {

Node\* temp = queue->front;

int data = temp->data;

queue->front = queue->front->next;

free(temp);

return data;

} else {

printf("Queue underflow!\n");

return -1;

}

}

int main() {

// Test stack using array

StackArray stackArray;

initStackArray(&stackArray);

pushStackArray(&stackArray, 1);

pushStackArray(&stackArray, 2);

pushStackArray(&stackArray, 3);

printf("Popped from stackArray: %d\n", popStackArray(&stackArray));

printf("Popped from stackArray: %d\n", popStackArray(&stackArray));

// Test queue using linked list

QueueLL queueLL;

initQueueLL(&queueLL);

enqueueLL(&queueLL, 1);

enqueueLL(&queueLL, 2);

enqueueLL(&queueLL, 3);

printf("Dequeued from queueLL: %d\n", dequeueLL(&queueLL));

printf("Dequeued from queueLL: %d\n", dequeueLL(&queueLL));

return 0;

}