#include <stdio.h>

#include <stdlib.h>

struct StackNode {

int data;

struct StackNode\* next;

};

struct Queue {

struct StackNode\* stack1;

struct StackNode\* stack2;

};

struct StackNode\* newStackNode(int data) {

struct StackNode\* node = (struct StackNode\*)malloc(sizeof(struct StackNode));

node->data = data;

node->next = NULL;

return node;

}

void push(struct StackNode\*\* top, int data) {

struct StackNode\* node = newStackNode(data);

node->next = \*top;

\*top = node;

}

int pop(struct StackNode\*\* top) {

if (\*top == NULL) {

printf("Stack underflow\n");

return -1;

}

struct StackNode\* temp = \*top;

\*top = (\*top)->next;

int popped = temp->data;

free(temp);

return popped;

}

int isEmpty(struct StackNode\* top) {

return top == NULL;

}

void initQueue(struct Queue\* queue) {

queue->stack1 = NULL;

queue->stack2 = NULL;

}

void enqueue(struct Queue\* queue, int data) {

push(&queue->stack1, data);

printf("Enqueued %d\n", data);

}

int dequeue(struct Queue\* queue) {

if (isEmpty(queue->stack1) && isEmpty(queue->stack2)) {

printf("Queue is empty\n");

return -1;

}

if (isEmpty(queue->stack2)) {

while (!isEmpty(queue->stack1)) {

int item = pop(&queue->stack1);

push(&queue->stack2, item);

}

}

int dequeuedValue = pop(&queue->stack2);

printf("Dequeued %d\n", dequeuedValue);

return dequeuedValue;

}

int main() {

struct Queue queue;

initQueue(&queue);

enqueue(&queue, 1);

enqueue(&queue, 2);

enqueue(&queue, 3);

dequeue(&queue);

dequeue(&queue);

enqueue(&queue, 4);

dequeue(&queue);

dequeue(&queue);

    return 0;

}