**Exercise 14: Write a program to implement Queue Sort**

#include <stdio.h>

#include <stdlib.h>

#define MAX 100

//there are 2 queues, and one empty linked list

//take elements from queues and sort it on linked list

struct q {

int queue[MAX];

int front;

int rear;

int counter;

};

struct node {

int x;

struct node \*next;

};

typedef struct node node;

typedef struct q queue;

void enqueue(queue \*\*queue, int x) {

if (!((\*queue)->counter == MAX)) {

(\*queue)->rear++;

(\*queue)->counter++;

if ((\*queue)->rear == MAX)

(\*queue)->rear = 0;

(\*queue)->queue[(\*queue)->rear] = x;

}

}

int dequeue(queue \*\*queue) {

if (!((\*queue)->counter == 0)) {

int x = (\*queue)->queue[(\*queue)->front];

(\*queue)->counter--;

(\*queue)->front++;

if ((\*queue)->front == MAX)

(\*queue)->front = 0;

return x;

}

else

return - 1;

}

node\* sortedInsert(node \*\*root, int x) {

if (\*root == NULL) {

\*root = (node \*)malloc(sizeof(node\*));

(\*root)->x = x;

(\*root)->next = NULL;

return root;

}

if (x< (\*root)->x) {

node \*temp = (node \*)malloc(sizeof(node));

temp->next = root;

temp->x = x;

return temp;

}

node \*iter = \*root;

while (iter->next != NULL && iter->next->x < x) {

iter = iter->next;

}

node \*temp = (node \*)malloc(sizeof(node));

temp->x = x;

temp->next = iter->next;

iter->next = temp;

return \*root;

}

void printlist(node \*root) {

while (root != NULL) {

printf("%d \n", root->x);

root = root->next;

}

}

void initialize(queue \*\*q) {

\*q = (queue\*)malloc(sizeof(queue));

(\*q)->counter = 0;

(\*q)->front = 0;

(\*q)->rear = -1;

}

void sortedlist(queue \*\*q1, queue \*\*q2, node \*\*root) {

while ((\*q1)->counter != 0) {

sortedInsert(root, dequeue(q1));

}

while ((\*q2)->counter != 0) {

sortedInsert(root, dequeue(q2));

}

}

int main(void) {

printf("hello\n");

queue \*q1 = NULL, \*q2 = NULL;

node \*root = NULL;

initialize(&q1);

initialize(&q2);

printf("init\n");

enqueue(&q1, 5);

enqueue(&q1, 20);

enqueue(&q1, 30);

enqueue(&q1, 40);

enqueue(&q2, 15);

enqueue(&q2, 19);

enqueue(&q2, 35);

enqueue(&q2, 65);

enqueue(&q2, 115);

printf("queued\n");

sortedlist(&q1, &q2, &root);

printf("sorted\n");

printlist(root);

}