Assignment 4

COMP 1123 - Data Structures I

Yaşar University

April 25, 2022

Multiple Queue Systems

For this assignment

- Design a structure, namely **MQUEUE**, to represent systems where multiple queues exist. MQUEUE structure should contain
  - **queues**: a linked queue array,
  - **queue_names** : names of queues,
  - **queue_count** : number of queues.
- Implement *mqueue_init()* function which initializes an MQUEUE with
  - an empty queue array,
  - an empty queue names array,
  - queue count zero.
- Implement *mqueue_add_queue_with_name(MQUEUE mqueue, char *queue_name)* function which adds a new queue to mqueue. It
  - initializes a new linked_queue and appends it to the **queues** array in mqueue, [1]
  - appends the *queue_name* to the **queue_names** array in mqueue, [1]
  - increments the **queue_count** in mqueue
- Impelement *mqueue_print(MQUEUE mqueue)* function which
  - prints the name of each queue in mqueue,
  - prints the elements of each queue in mqueue.
- Implement *mqueue_enqueue_with_name(MQUEUE mqueue, char *queue_name, void *data)* function which enqueues data to the queue with given name in mqueue.
- Implement *mqueue_dequeue_with_name(MQUEUE mqueue, char *queue_name)* function which dequeues from the queue with given name in mqueue.
- Implement *mqueue_free(MQUEUE mqueue)* function which frees the mqueue.

[1] Here you will need to increase the size of the array. To do that, you may use the *realloc* function which is described below.

Main

We provide you a main code to simulate Airport Passport Control Queues. Usually there are two queues for passport control: EU-Citizen Queue and Non-EU Citizen Queue. In main, we create an MQUEUE and add two queues for EU and Non-EU citizens. We enqueue citizen numbers to the queues and dequeue them afterwards, randomly. At the end, we print the MQUEUE.

```
int main() {
    srand(time(NULL));
    MQUEUE mqueue = mqueue_init();
    mqueue_add_queue_with_name(mqueue, "EU Passport Control");
    mqueue_add_queue_with_name(mqueue, "Non-EU Passport Control");

    int rnd, *data;
    char *names[] = {"EU Passport Control", "Non-EU Passport Control"};

    for(int i = 0; i < 30; i++) {
        rnd = rand() % 100;
        data = (int *)malloc(sizeof(int));
        *data = rand() % 100;
        if(rnd < 20) {
            mqueue_enqueue_with_name(mqueue, "EU Passport Control", data);
            printf("Citizen %d\t enters to \tEU Passport Control\n", *data);
        } else if(rnd < 60) {
            mqueue_enqueue_with_name(mqueue, "Non-EU Passport Control", data);
            printf("Citizen %d\t enters to \tNon-EU Passport Control\n", *data);
        }
        rnd = rand() % 100;
        if(rnd < 30) {
            data = mqueue_dequeue_with_name(mqueue, "EU Passport Control");
            if(data != NULL) {
                printf("Citizen %d\t exits \t\tEU Passport Control\n", *data);
            }
        } else if(rnd < 50) {
            data = mqueue_dequeue_with_name(mqueue, "Non-EU Passport Control");
            if(data != NULL) {
                printf("Citizen %d\t exits \t\tNon-EU Passport Control\n", *data);
            }
        }
    }
    printf("\n---------------------------\n");
    printf("\nCurrent queue lines:\n\n");
    mqueue_print(mqueue);

    mqueue_free(mqueue);

    return 0;
}
```

You should implement all the functions to be able to run the main code. Except the randomized information, your output should be similar to the example output given below.

```
Citizen 96       enters to      Non-EU Passport Control
Citizen 96       exits          Non-EU Passport Control
Citizen 39       enters to      EU Passport Control
Citizen 81       enters to      Non-EU Passport Control
Citizen 60       enters to      EU Passport Control
Citizen 39       exits          EU Passport Control
Citizen 45       enters to      Non-EU Passport Control
Citizen 60       exits          EU Passport Control
Citizen 36       enters to      Non-EU Passport Control
Citizen 81       exits          Non-EU Passport Control
Citizen 28       enters to      Non-EU Passport Control
Citizen 41       enters to      Non-EU Passport Control
Citizen 45       exits          Non-EU Passport Control
Citizen 6        enters to      Non-EU Passport Control
Citizen 64       enters to      EU Passport Control
```

```
Citizen 64       exits          EU Passport Control
Citizen 70       enters to      Non-EU Passport Control
Citizen 25       enters to      Non-EU Passport Control
Citizen 24       enters to      Non-EU Passport Control
Citizen 36       exits          Non-EU Passport Control
Citizen 61       enters to      EU Passport Control
Citizen 61       exits          EU Passport Control
Citizen 31       enters to      Non-EU Passport Control
Citizen 43       enters to      EU Passport Control
Citizen 43       exits          EU Passport Control
Citizen 28       exits          Non-EU Passport Control
Citizen 10       enters to      EU Passport Control
Citizen 40       enters to      Non-EU Passport Control
Citizen 41       exits          Non-EU Passport Control
Citizen 46       enters to      Non-EU Passport Control

---------------------------

Current queue lines:

QUEUE NAME: EU Passport Control
QUEUE: 10
-------
QUEUE NAME: Non-EU Passport Control
QUEUE: 6 70 25 24 31 40 46
-------

```

Hints

The size of the **queues/queue_names** array of an mqueue should change as you add new queues. If there are no queues in mqueue, you should not allocate memory for queues or queue names. If you add a new queue, the size of the array should increase. To do that you may need the *realloc* function which changes the size of memory allocated for an adrress. An example is as follows.

```
int *arr = (int *)malloc(sizeof(int) * 5);
arr = realloc(arr, sizeof(int) * 10);
// increased the size of arr from 5 to 10
```

Notes

- Do not use any library other than *stdio.h*, *stdlib.h*, *time.h* and *string.h*.
- Any kind of cheating (over the internet, between students, etc.) is not allowed.
- Upload a single .c file.