English version | [ver versão em português]

Versão em Português | [see english version]

[PI057] Two Queues

In this problem, you should only submit a file **pi057.c** containing a line with `#include "queue.h"` and a function **process()** as described below (a complete program is not required). See the code to test at the end.

You may assume that the "queue implementation" used in class will be available on Mooshak.

Method to Submit

- **`void process(Queue *q, Queue *a, Queue *b)`**

This function should process the elements in the queue `q`, where each pair consists of a name followed by an operation. Depending on the operation, the method should:

- **Name A** - Add the name to queue `a`
- **Name B** - Add the name to queue `b`
- **Name X** - Add the name to the queue with fewer elements (`a` or `b`). If both queues have the same number of elements, the name is discarded and not added to either.

For example, if the queue `q` is {Luis,B,Pedro,A,Luisa,A,Joao,X,Jose,X,Miguel,B}, the following happens:

1. *Luis B* - Luis is added to queue `b`
2. *Pedro A* - Pedro is added to queue `a`
3. *Luisa A* - Luisa is added to queue `a`
4. *Joao X* - Joao is added to queue `b`, which has only 1 element (Luis) compared to queue `a` (Pedro and Luisa)
5. *Jose X* - Jose is discarded (both queues have 2 elements)
6. *Miguel B* - Miguel is added to queue `b`

At the end, queue `a` contains {Pedro, Luisa} and queue `b` contains {Luis, Joao, Miguel}. Queue `q` should be empty.

Examples of input/output

Queue *q*, *a* eand *b* at start	Queue *q*, *a* and *b* after the call
q = {Luis,B,Pedro,A,Luisa,A,Joao,X,Jose,X,Miguel,B} a = {} b = {}	q = {} a = {Pedro, Luisa} b = {Luis, Joao, Miguel}
q = {Luis,B,Pedro,B,Luisa,X,Joao,X} a = {} b = {}	q = {} a = {Luisa,Joao} b = {Luis,Pedro}

[PI057] Duas filas

Neste problema deverá apenas submeter um ficheiro **pi057.c** contendo uma linha com `#include "queue.h"` e uma função **process()** como a seguir se descreve (não é necessário um programa completo). Ver o código para fazer testes no final.

Pode assumir que terá acesso no Mooshak à "implementação de filas" como dadas nas aulas.

Método a submeter

- **`void process(Queue *q, Queue *a, Queue *b)`**

Esta função deve processar os elementos que vêm na fila `q` na forma de um nome seguido de uma operação. Consoante a operação deve fazer o seguinte:

- **Nome A** - Adiciona nome à fila `a`
- **Nome B** - Adiciona nome à fila `b`
- **Nome X** - Adiciona nome à fila que tenha menos elementos (`a` ou `b`). Se ambas as filas tiverem o mesmo número de elementos, o nome é descartado e não é adicionada a nenhuma.

Por exemplo, se a fila `q` fosse {Luis,B,Pedro,A,Luisa,A,Joao,X,Jose,X,Miguel,B} acontecia o seguinte:

1. *Luis B* - Luis é adicionado à fila `b`
2. *Pedro A* - Pedro é adicionado à fila `a`
3. *Luisa A* - Luisa é adicionada à fila `a`
4. *Joao X* - Joao é adicionado à fila `b`, que tem apenas 1 elemento (Luis) contra os dois da fila `b` (Pedro e Luisa)
5. *Jose X* - José descartado (ambas as filas têm 2 elementos)
6. *Miguel B* - Miguel é adicionado à fila `b`

No final a fila `a` fica com {Pedro,Luisa} e a fila `b` fica com {Luis,Joao,Miguel}. A fila `q` deve ficar vazia.

Exemplos de input/output

Fila *q*, *a* e *b* no início	Fila *q*, *a* e *b* no final da chamada
q = {Luis,B,Pedro,A,Luisa,A,Joao,X,Jose,X,Miguel,B} a = {} b = {}	q = {} a = {Pedro, Luisa} b = {Luis, Joao, Miguel}
q = {Luis,B,Pedro,B,Luisa,X,Joao,X} a = {} b = {}	q = {} a = {Luisa,Joao} b = {Luis,Pedro}

Como compilar localmente

Para compilar localmente vai precisar de ter os ficheiros seguintes:

- queue.h e queue.c -- implementação de uma fila. Verificar se o NodeInfo está configurado para o tipo de dados a guardar na fila.
- pi057.c -- o seu código com a função pedida
- tests057.c -- ficheiro com a função main() e onde se trata da leitura e escrita.

Para compilar execute na pasta onde estão os ficheiros: `gcc tests057.c queue.c pi057.c` que vai gerar um executável `a.out`

Código para testes, ficheiro: `tests057.c`

```c
// PI057 Testing Two Queues Problem
//

#include "queue.h"

void process(Queue *, Queue *, Queue *);

int main() {

  Queue q, a, b;
  initQueue(&q);   // queue with input
  initQueue(&a);   // result queue for A
  initQueue(&b);   // result queue for B

  int n;

  scanf("%d", &n);                  // number of pairs
  for (int i=0; i<n; i++) {
    char *s= (char *)malloc(sizeof(MAXCHARS));
    char *op= (char *)malloc(sizeof(2));
    scanf("%s %s", s, op);
    addLast(&q,s);
    addLast(&q,op);
  }

  printf("Inicio: ");
  printf("q: "); printQueue(&q); printf("\n");
  printf("a: "); printQueue(&a); printf("\n");
  printf("b: "); printQueue(&b); printf("\n");
  process(&q, &a, &b);
  printf("------------------------------\n");
  printf("Final: ");
```

```
  printf("q: "); printQueue(&q); printf("\n");
  printf("a: "); printQueue(&a); printf("\n");
  printf("b: "); printQueue(&b); printf("\n");
}
```
