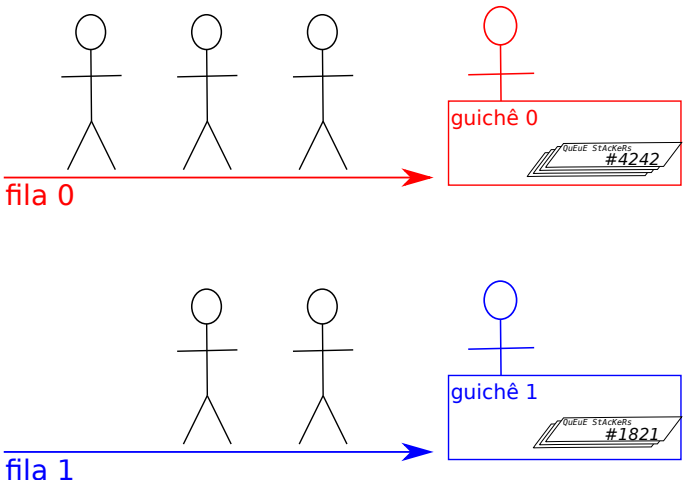
Algoritmos e Estruturas de Dados 1

2024/1 - Trabalho da Disciplina

Enunciado

A famosa banda internacional `QuEuE StAcKeRs` finalmente virá ao Brasil, e fará um grande show em Toledo! Como a procura por ingressos está muito grande, a organização pediu sua ajuda para gerenciar a bilheteria do evento.

A bilheteria contém dois guichês: o guichê 0 e o guichê 1. Cada guichê mantém uma pilha de ingressos, e cada ingresso é identificado por um número de série. Além disso, há duas filas: a fila 0 e a fila 1, que podem comportar até 20 pessoas simultaneamente cada uma. O guichê 0 atende apenas clientes da fila 0, e o guichê 1 atende apenas clientes da fila 1.



Inicialmente, tanto as filas quanto as pilhas de ingressos estão vazias. Escreva um programa em C que lê uma sequência de comandos do usuário e os processa de acordo. Os possíveis comandos são:

- `chega` P N: uma pessoa cujo nome é P e que deseja comprar N ingressos chega na bilheteria. Como exemplo, o comando `chega Ana 3` indica que Ana chegou na bilheteria, e que ela deseja comprar 3 ingressos.

  Ao chegar, toda pessoa entra no final da fila de menor tamanho no momento. Se ambas as filas têm o mesmo tamanho no momento, a pessoa entra na fila 0. Entretanto, se ambas as filas estão cheias no momento, a pessoa desiste e vai embora.

  Para cada comando deste tipo, imprima uma linha contendo "P `entrou na fila` F`.`", onde P é o nome da pessoa, e F é a fila na qual ela entrou (0 ou 1), ou "P `desistiu, filas cheias.`" se ambas as filas estão cheias no momento.

- `carrega` G A B: empilha, na pilha do guichê G, todos os ingressos cujos números de série estão entre A e B, inclusive (de forma que o ingresso de número B fica no topo da pilha). Como exemplo, o comando `carrega 0 4240 4242` empilha, nesta ordem, os ingressos 4240, 4241 e 4242 na pilha do guichê 0.

  Para cada comando deste tipo, imprima uma linha contendo "N `ingresso(s) carregado(s) no guiche` G`.`", onde N é o número de ingressos carregados no guichê G.

- `vende` G: o guichê G faz uma venda. Como exemplo, o comando `vende 1` indica que o guichê 1 faz uma venda.

  Se a fila G estiver vazia, imprima "`Fila` G `vazia!`" e ignore o comando. Caso contrário, a pessoa no início da fila G compra todos os ingressos que deseja, na ordem em que são desempilhados do guichê. Para cada ingresso comprado, imprima "P `comprou ingresso #`I`.`", onde P é o nome da pessoa e I é o número do ingresso comprado.

  Se o guichê tiver menos ingressos disponíveis que a quantidade desejada pela pessoa, a pessoa compra todos os ingressos disponíveis no guichê no momento. Entretanto, se o guichê não tiver *nenhum* ingresso disponível, imprima "`Guiche` G `sem ingressos!` P `triste.`".

  Após o comando, a pessoa P sai da bilheteria.

- `fim`: encerra o programa. Ao final, imprima "`Ingressos vendidos:` T`.`", onde T é o número total de ingressos vendidos.

Você pode assumir que serão carregados nos guichês no máximo dois mil ingressos ao todo. Além disso, todo nome terá no máximo 20 caracteres e não conterá espaços.

Exemplo de entrada	Exemplo de saída
chega Ana 3 vende 1 chega Joao 2 vende 1 carrega 0 4240 4242 vende 0 carrega 1 1815 1821 chega Marcinha 1 chega Daniel 3 chega Nilo 2 carrega 0 1 1 carrega 0 32 33 vende 0 vende 0 vende 1 chega Andre 1 vende 0 vende 0 fim	Ana entrou na fila 0. Fila 1 vazia! Joao entrou na fila 1. Guiche 1 sem ingressos! Joao triste. 3 ingresso(s) carregado(s) no guiche 0. Ana comprou ingresso #4242. Ana comprou ingresso #4241. Ana comprou ingresso #4240. 7 ingresso(s) carregado(s) no guiche 1. Marcinha entrou na fila 0. Daniel entrou na fila 1. Nilo entrou na fila 0. 1 ingresso(s) carregado(s) no guiche 0. 2 ingresso(s) carregado(s) no guiche 0. Marcinha comprou ingresso #33. Nilo comprou ingresso #32. Nilo comprou ingresso #1. Daniel comprou ingresso #1821. Daniel comprou ingresso #1820. Daniel comprou ingresso #1819. Andre entrou na fila 0. Guiche 0 sem ingressos! Andre triste. Fila 0 vazia! Ingressos vendidos: 9.

Exemplo de entrada	Exemplo de saída
chega Pessoa1 1 chega Pessoa2 1 chega Pessoa3 1 ... chega Pessoa39 1 chega Pessoa40 1 chega Pessoa41 1 chega Pessoa42 1 fim	Pessoa1 entrou na fila 0. Pessoa2 entrou na fila 1. Pessoa3 entrou na fila 0. ... Pessoa39 entrou na fila 0. Pessoa40 entrou na fila 1. Pessoa41 desistiu, filas cheias. Pessoa42 desistiu, filas cheias. Ingressos vendidos: 0.

Implementação

O trabalho deve **obrigatoriamente** usar pilha(s) **e** fila(s) em sua solução. O trabalho deve conter os seguintes arquivos:

- `PE.h` e `PE.c`: definição e implementação de pilha usando como base um vetor ("pilha estática");
- `PD.h` e `PD.c`: definição e implementação de pilha usando como base uma lista ligada ("pilha dinâmica");
- `FE.h` e `FE.c`: definição e implementação de fila usando como base um vetor ("fila estática");
- `FD.h` e `FD.c`: definição e implementação de fila usando como base uma lista ligada ("fila dinâmica");
- `main.c`: programa principal. Deve incluir (via `#include`):
  - `PE.h` *ou* `PD.h`; *e*
  - `FE.h` *ou* `FD.h`.

O programa principal deve utilizar filas e pilhas como estruturas *abstratas* de dados. Em particular, deve ser possível "escolher" entre usar pilhas estáticas ou dinâmicas *apenas alterando os* `#include` *e recompilando de acordo*! Da mesma forma, deve ser possível "escolher" entre usar filas estáticas ou dinâmicas de maneira análoga (note que, desta forma, há um total de quatro "configurações" com as quais o trabalho deverá funcionar).

Independentemente da implementação, certifique-se que toda memória alocada por seu programa é desalocada ao final da sua execução.

Orientações

- O trabalho pode ser feito por equipes de *até* 2 (dois) estudantes;
- Submeta, via *Moodle*, um pacote (zip ou tar.gz) contendo os 9 arquivos citados acima, além de um arquivo de texto (txt) onde conste:
  - O nome de todos os integrantes da equipe;
  - Toda informação que a equipe julgar relevante para a correção (como *bugs* conhecidos, detalhes de implementação, escolhas de projeto, etc.)
- Comente adequadamente seus códigos para facilitar a correção.
- Atenção: a correção será parcialmente automatizada, e a saída do programa será testada com outras entradas além das fornecidas como exemplo. *Siga* ***fielmente*** *o formato de saída dado nos exemplos*, sob pena de grande redução da nota;
- Certifique-se que seu programa funciona antes de submetê-lo;
- O trabalho deve ser entregue até **26 de Maio de 2024, 23:59**, via *Moodle*. É suficiente que o trabalho seja submetido por apenas um estudante da equipe;
- Trabalhos detectados como cópia/plágio (de colegas, da internet ou de ferramentas de IA), ou comprados, receberão **todos** a nota 0 (**ZERO**) e estarão sujeitos a abertura de Processo Administrativo Disciplinar Discente.