

Trabalho 1

O trabalho 1 é uma continuação do ex2. Ou seja, os .c e .h podem ser reaproveitados do ex2 (com algumas modificações).

Neste trabalho o programador deve criar um catálogo como **LISTA CIRCULAR DUPLAMENTE ENCADEADA**.

O Trabalho 1 pode ser feito em dupla. Porém deve haver um documento com o nome e número USP dos integrantes da dupla.

Nota: O CSV.csv está em UTF-8, logo há 3 bytes no início do arquivo “informando” isso. Para ler os dados o aluno não deve considerar esses 3 primeiros bytes. Uma das formas de desconsiderar esses 3 bytes é usando “fseek”.

Objetivo

O objetivo deste exercício prático é estimular os estudantes a se familiarizarem com o comportamento e a lógica associados à estrutura de dados **LISTA CIRCULAR DUPLAMENTE ENCADEADA**.

Queremos que os alunos se habituem com a implementação dessa estrutura e consigam entender as funções básicas, que garantem a execução adequada desse TAD, em diferentes contextos.

Para acostumar os alunos com conceitos de modularização e boas práticas de escrita de código, será exigido o uso de múltiplos arquivos .c e .h no projeto, bem como a construção de um arquivo Makefile, responsável por gerenciar a execução do programa.

Descrição

O Programa deve receber como entrada um **CSV** com os seguintes dados:

Nome do jogo (coluna 1), ano de lançamento (coluna 2), produtora (coluna 3); sendo cada linha um novo jogo. TODOS os jogos do CSV devem ser adicionados ao catálogo.

O retorno do programa é baseado no input do USUARIO da seguinte forma:

- Caso o usuário entre com a letra “**r**”, verifica e remove possíveis jogos duplicados no catálogo.
- Caso o usuário entre com a letra “**p**” (seguido “produtora” - nome da produtora como input), retorna todos os jogos da produtora – mesma função do ex2.
- Caso o usuário entre com a letra “**a**” (seguido por “ano”), retorna todos os jogos lançados no ano desejado pelo usuário.
- Caso o usuário entre com a letra “**i**”, imprime o nome de todos os jogos conforme a ordem do catálogo.
- Caso o usuário entre com a letra “**u**” (seguido por um inteiro “Int”), imprime o jogo na posição “int” do catálogo.
- Caso o usuário entre com a letra “**mr**” (seguido por dois inteiros “int 1” e “int 2”), o programa deve mover “int 2” casas para a direita o jogo em “int 1”. Ex: mr 3 6 -> o usuário deve mover 6 casas para a direita o jogo que está na casa 3.
- Caso o usuário entre com a letra “**ml**” (seguido por dois inteiros “int 1” e “int 2”), o programa deve mover “int 2” casas para a esquerda o jogo em “int 1”. Ex: mr 3 6 -> o usuário deve mover 6 casas para a esquerda o jogo que está na casa 3.
- Caso o usuário entre com a letra “**f**”, o programa deve encerrar e desalocar toda a memória alocada.

Nota: Como a lista é circular as operações “**mr**” e “**ml**” não tem limite de casas para a movimentação.

Exemplos:

CSV de entrada (para todos os exemplos abaixo):

	A	B	C	D
1	Assassin's Creed	2007	Ubisoft	
2	Assassin's Creed IV: Black Flag	2013	Ubisoft	
3	DmC: Devil May Cry	2013	Ninja Theory	
4	Dragon Quest VII	2000	Square Enix	
5	Fable	2004	Lionhead Studios	
6	Assassin's Creed IV: Black Flag	2013	Ubisoft	
7				

Exemplo 1:

Entrada:

p Ubisoft

p Square Enix

f

Saida:

Assassin's Creed

Assassin's Creed IV: Black Flag

Assassin's Creed IV: Black Flag

Dragon Quest VII

Exemplo 2:

Entrada:

a 2013

r

a 2004

a 2013

f

Saida:

Assassin's Creed IV: Black Flag

DmC: Devil May Cry

Assassin's Creed IV: Black Flag

Fable

Assassin's Creed IV: Black Flag

DmC: Devil May Cry

Exemplo 3:

Entrada:

i

mr 3 1

i

mr 0 8

i

f

Saida:

Assassin's Creed

Assassin's Creed IV: Black Flag

DmC: Devil May Cry

Dragon Quest VII

Fable

Assassin's Creed IV: Black Flag

Assassin's Creed

Assassin's Creed IV: Black Flag

DmC: Devil May Cry

Fable

Dragon Quest VII

Assassin's Creed IV: Black Flag

Assassin's Creed IV: Black Flag

DmC: Devil May Cry

Assassin's Creed

Fable

Dragon Quest VII

Assassin's Creed IV: Black Flag

Exemplo 4:

Entrada:

u 0

u 6

u 7

f

Saida:

Assassin's Creed

Assassin's Creed

Assassin's Creed IV: Black Flag

Observações da avaliação

A avaliação do seu programa será feita além do resultado da plataforma run.codes. Portanto, ter um bom resultado com os casos de teste, não será suficiente para garantir a nota máxima e nem a aprovação do exercício.

Caso seu projeto não satisfaça os pontos exigidos nos objetivos e explicitados nas observações de implementação, sua nota poderá ser reduzida ou ser desconsiderada.

Cópias de código entre alunos, acusadas pela plataforma, resultarão imediatamente em zero aos dois ou mais alunos envolvidos.