

MAC 323 – Estruturas de Dados**Primeiro semestre de 2005****Exercício-Programa 1 – Entrega: 5 de abril de 2005**

Jogo de paciência

Neste exercício-programa vocês deverão simular um jogo de paciência que descrevemos a seguir. O jogo é jogado com apenas um baralho (52 cartas). As quatro primeiras cartas de paus (ás, 2, 3 e 4) são retiradas e colocadas abertas nas pilhas de saída. As 48 cartas restantes são embaralhadas e deixadas com as faces para baixo na pilha de entrada. Além das pilhas de saída, existem 4 pilhas de jogo cuja função descrevemos a seguir.

As pilhas de saída aceitam cartas de qualquer naipe, e em ordem crescente. A primeira pilha (a do ás) cresce de um em um. A segunda, de dois em dois. A terceira de três em três e a quarta de quatro em quatro. Assim, sobre o dois de paus (segunda pilha) você pode colocar, por exemplo, o 4 de ouros. Sobre o 4 de paus (quarta pilha) você pode colocar o oito de espadas. O objetivo do jogo é, no final, ter todas as 52 cartas nas pilhas de saída, que terão quatro reis nos topos. As contas são feitas circularmente. Assim, sobre uma dama na terceira pilha poderá ser colocado um 2.

Eventualmente uma carta da pilha de entrada não poderá ser colocada imediatamente em qualquer uma das pilhas de saída. Para isso servem as pilhas de jogo. Esta carta poderá ser colocada em qualquer uma delas. Os topos das pilhas de jogo podem ser movidos, quando possível, para alguma pilha de saída. A estratégia do jogo consiste em manipular bem essas pilhas de jogo.

Cartas da pilha de entrada que servem em alguma pilha de saída podem, ainda assim, ir para uma pilha de jogo, se você achar que isso é uma boa estratégia.

Quando todas as cartas da pilha de entrada forem utilizadas, e nenhuma das cartas dos topos das pilhas de jogos puderem ser movimentadas para pilhas de saída o jogo termina.

Faça um programa em C ou Java que simule um jogo de paciência como descrito acima, permitindo que diversos jogos possam ser executados. Para vocês que não conhecem jogos de cartas, as cartas de um baralho estão divididas em 4 naipes (ouros \spadesuit , copas \heartsuit , paus \clubsuit e espadas \diamondsuit). Para cada naipe existem 13 cartas: A (ás – 1), 2, 3, 4, 5, 6, 7, 8, 9, 10, J (valete – 11), Q (dama – 12) e K (rei – 13). Escolha uma implementação mais eficiente possível para armazenar as cartas.

Você deverá escrever métodos (funções) para manipular as várias pilhas do seu programa (entrada, saída e jogo), que **deverão ser armazenadas todas no mesmo vetor, com 55 posições**. Note que ao mover as cartas de uma pilha para outra ocorrerão colisões e serão necessárias movimentações das pilhas no vetor. A estrutura deve ser econômica e ao mesmo tempo deve dar um tratamento eficiente ao problema. Para você se inspirar, no livro “The art of Computer Programming”, Vol I, de Don Knuth, ele trata deste problema.

Se você desejar, implemente uma versão que permite ao usuário jogar *on line* o jogo (bônus na nota para os que fizerem).