MAC–IME–USP

Carlos Eduardo Ferreira
Sala 297A Tel.: 818 6140
e-mail cef@ime.usp.br

MAC 323 – Estruturas de Dados

Primeiro semestre de 1998

Exercício-Programa 1 – Entrega: **24 de março de 1998**

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++, Delphi ou Java), que simule um jogo de paciência como descrito acima, permitindo que diversos jogos possam ser executados. Para vocês que não conhecem jogos, as cartas de um baralho estão divididas em 4 naipes (ouros ♦, copas ♥, paus ♣ e espadas ♠). 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.

TODAS AS PILHAS NECESSÁRIAS PARA SEU PROGRAMA DEVEM SER ARMAZENADAS EM UM ÚNICO VETOR COM 60 POSIÇÕES, e para o tratamento de overflow você deve usar o algoritmo de realocação de tabelas seqüenciais, descrito no livro do Knuth, vol. I, pags. 245 e 246. Observe que, devido ao tamanho máximo do vetor, ocorrerão muitas colisões entre as pilhas, portanto, seu tratamento deve ser o mais eficiente possível.

Você deve entregar um disquete contendo o programa fonte e uma listagem do programa. Seu programa deve imprimir o conteúdo das pilhas a cada jogada, assim como uma mensagem dizendo que movimento foi realizado. As pilhas de jogo DEVERÃO ser impressas em forma de pilhas (uma ao lado da outra), e lembre-se de diferenciar as cartas abertas das fechadas. Obviamente seu programa deverá ser bem documentado, e a forma de utilizá-lo também será avaliada.

Quem quiser treinar um pouco do jogo antes de começar a programar poderá testar a versão para Windows disponível na página do curso:

`http://www.ime.usp.br/~cef/mac323`

Se você desejar, implemente uma versão que permite ao usuário jogar *on line* o jogo (bônus na nota para os que fizerem). As melhores estratégias de jogo serão recompensadas na nota!!