Universidade Federal do Estado de Minas Gerais
Professor: Edwaldo Soares Rodrigues
AEDS II

Trabalho Prático 1

Instruções:

I. O trabalho deverá ser feito em dupla.
II. O trabalho deverá ser realizado usando a linguagem de programação C.
III. Deverão usar os conceitos aprendidos na disciplina de AEDS II, levando em consideração as melhores estruturas para representar os itens do jogo (baralho, mesa, e as mãos dos jogadores).
IV. O trabalho deverá ser entregue até a data 09/12/2018.
V. O trabalho deverá ser enviado até na data limite no seguinte e-mail: edwaldoroadsf1@yahoo.com.br
VI. O assunto do e-mail deverá seguir o padrão: TP_AEDSII_NomeAluno1_NomeAluno2

Enunciado do Trabalho:

O Trabalho Prático 1 deverá ser implementado usando a linguagem de programação C e tem como objetivo colocar em prática os conceitos vistos acerca de Algoritmos e Estruturas de Dados II ao longo do semestre.

Problema – Rouba Montes:

Um dos jogos de cartas mais divertidos para crianças pequenas, pela simplicidade, é o Rouba- Monte. O jogo utiliza um ou mais baralhos normais e tem regras muito simples. Cartas são distinguidas apenas pelo valor (ás, dois, três, . . .), ou seja, os naipes das cartas não são considerados (por exemplo, ás de paus e ás de ouro têm o mesmo valor).

Inicialmente as cartas são embaralhadas e colocadas em uma pilha na mesa de jogo, chamada de pilha de compra, com face voltada para baixo. Durante o jogo, cada jogador mantém um monte de cartas, com face voltada para cima; em um dado momento o monte de um jogador pode conter zero ou mais cartas. No início do jogo, todos os montes dos jogadores têm zero cartas. Ao lado da pilha de compras encontra-se uma área denominada de área de descarte, inicialmente vazia, e todas as cartas colocadas na área de descarte são colocadas lado a lado com a face para cima (ou seja, não são empilhadas).

Os jogadores, dispostos em um círculo ao redor da mesa de jogo, jogam em sequência, em sentido horário. As jogadas prosseguem da seguinte forma:

- O jogador que tem a vez de jogar retira a carta de cima da pilha de compras e a mostra aos outros jogadores; vamos chamar essa carta de carta da vez.

- Se a carta da vez for igual a alguma carta presente na área de descarte, o jogador retira essa carta da área de descarte colocando-a, juntamente com a carta da vez, no topo de seu monte, com as faces voltada para cima, e continua a jogada (ou seja,

retira outra carta da pilha de compras e repete o processo).

- Se a carta da vez for igual à carta de cima de um monte de um outro jogador, o jogador "rouba"esse monte, empilhando-o em seu próprio monte, coloca a carta da vez no topo do seu monte, face para cima, e continua a jogada.

- Se a carta da vez for igual à carta no topo de seu próprio monte, o jogador coloca a carta da vez no topo de seu próprio monte, com a face para cima, e continua a jogada.

- Se a carta da vez for diferente das cartas da área de descarte e das cartas nos topos dos montes, o jogador a coloca na área de descarte, face para cima, e a jogada se encerra (ou seja, o próximo jogador efetua a sua jogada). Note que esse é o único caso em que o jogador não continua a jogada.

O jogo termina quando não há mais cartas na pilha de compras. O jogador que tiver o maior monte (o monte contendo o maior número de cartas) ganha o jogo. Se houver empate, todos os jogadores com o monte contendo o maior número de cartas ganham o jogo.

Metodologia:

Deve ser incluída uma struct para representar uma Carta, esta deve ter dois membros, número e nipe. Observe que o membro nipe não será utilizado mas deve estar presente na struct. O baralho, a mesa e as mãos dos jogadores devem ser compostos por Cartas.

Seu código deve permitir que se escolha a quantidade de jogadores que vão jogar, bem como a quantidade de cartas que poderão ser usadas no jogo. Em seguida, após a execução do jogo, o código deve apresentar o(s) ganhador(es) (nome do jogador e número de cartas em mãos), além de apresentar de forma ordenada as cartas presentes na mão do ganhador(es).

Critérios de avaliação:

- Escolha das estruturas de dados para representação dos itens do jogo;
- Código legível, identado e bem comentado;
- Variáveis, structs e funções com nomes representativos;
- Uso adequado dos recursos de linguagem;
- Modularização e organização do código;