PARTE 1 – CAIXA ELETRÔNICO

Faça um programa que simule um caixa eletrônico, para isso, desenvolva:

Struct para Cadastro de Contas contendo, no mínimo: Nome do Titular, Número da Conta, Saldo Bancário;

Restrições:

- O número da conta deve ser gerado automaticamente pelo sistema, através de uma função específica, contendo 3 dígitos, cujo valor obrigatoriamente é múltiplo de 7 e que seja único para cada conta criada.
- Saldo inicial sempre é zero.

Struct para Cadastro de Movimentação Bancária contendo, no mínimo: Tipo de Movimentação; Descrição da Movimentação; Conta de Origem; Conta de Destino; Valor da Movimentação.

Restrições:

- Tipo de movimentação pode assumir apenas dois valores possíveis, ou débito ou crédito;
- Descrição é um valor textual informado pelo usuário durante a movimentação;
- Para operações de Saque, conta de destino será um valor nulo (0);
- Para operações de Deposito, conta de origem será um valor nulo (0);

Além das Structs, o programa deve implementar as seguintes operações básicas, utilizando funções específicas para tratamento dessas operações. (Pode-se implementar outras funções que facilitem o desenvolvimento da solução, a critério do time).

1- Cadastro de Conta
2- Depósito
3- Saque
4- Transferência entre contas
5- Registro Geral de Movimentações
6- Relatório de Contas (ordenado de forma decrescente com base no valor do saldo);

Restrições:

- Ao realizar Depósito,Saque ou Transferência, o programa deve informar se o número da conta é inválido (ou seja, não corresponde ao protocolo de criação de conta) ou se o número da conta não existe.
- Dica Útil: Criar uma função específica para buscar e retornar o índice da localização da conta no repositório de Contas, ou -1 caso ela não exista.
- O programa não deve permitir uma operação que torne o saldo de uma conta negativa;
- Ao final de cada operação, o programa deve retornar a tela inicial do programa, apresentando todas as opções novamente.

PARTE 2 - JOGO DO BURRO

Um baralho contem 52 Cartas.

Cada Carta é pode ser representada por uma *struct* contendo, no mínimo, informações de:

- Valor (1 a 13),
- Naipe (#, @, &, $)
- Situação, onde:
    -1 = Carta fora de jogo;
    0 = Carta no monte;
    1 = Carta em posse do Jogador 1 (Usuário);
    2 = Carta em posse do Jogador 2 (Computador);

Faça um programa que simule um "Jogo de Burro" simplificado, conforme as regras a seguir...

* Neste jogo, o usuário tem o próprio computador como adversário.

* Os 2 jogadores (usuário e PC) recebem 4 cartas aleatórias;

* O usuário sempre inicia a primeira rodada do jogo, escolhendo uma carta para jogar.

* A lógica do jogo é a seguinte, o oponente (seja computador ou usuário) sempre deve jogar uma carta com o mesmo naipe da primeira carta da rodada.

* Caso não possua uma carta do mesmo naipe, o jogador (usuário ou PC) deve “comprar” cartas aleatórias do monte do baralho, até que encontre uma carta do mesmo naipe da rodada.

* Caso todas as cartas do monte acabem, o jogador (usuário ou PC) perde automaticamente.

* Em cada rodada, o jogador (usuário ou PC) com a carta de maior valor na mesa vence a rodada, e as duas cartas então são descartadas (tornam-se fora de jogo).

* O jogo recomeça sempre pelo ganhador da rodada anterior.

Você deve implementar a solução deste problema com uma interface que possibilite entender e acompanhar o jogo do início ao fim.