

```

#include"stdio.h"
#include"stdlib.h"
#include"string.h"
#include"locale.h"
#define NULO '\0'
leituraArquivo(char conta[], int linha)
{
//função para ler as linhas dentro do arquivo.
FILE *arq;
char c[7];
int i, result;
float x=0;
arq = fopen(conta, "r+");// Abre um arquivo TEXTO para LEITURA
if (arq == NULL)// Se houve erro na abertura apresenta a mensagem abaixo.
{
printf("Problemas na abertura do arquivo\n");//mensagem.
return;
}
i = 1;//essa variável contém o valor 1 para começar apartir da primeira linha dentro do arquivo com o código
abaixo
while (!feof(arq))
{
result = fscanf(arq, "%s %f",&c ,&x);//Lê uma linha dentro do arquivo(inclusive com o '\n')
if(i == 1 && linha == 1)
{
if (i == linha)
{//função que verifica se a variável i for igual a variável linha.Código abaixo é semelhante e faz a
mesma função.
return x;//retorna a variável x
break;//linha para parar o ciclo.
}
}
if(i == 2 && linha == 2)
{
if (i == linha)
{
return x;
break;
}
}
if(i == 3 && linha == 3)
{
if (i == linha)
{
return x;
break;
}
}
if(i == 4 && linha == 4)
{
if (i == linha)
{
return x;
break;
}
}

i++;//++ para acrescentar +1 na variável i
}
fclose(arq);//fecha o arquivo
}
gravarArquivo(char conta[], int linhaArq, float linhaValor)
{
//função para gravar linhas dentro do arquivo
FILE *arqEntrada;//ponteiro para a variável arqEntrada
FILE *arqSaida;//ponteiro para a variável arqSaida
int linha = 1;
int i,limite;

```

```

char textoArquivo[100];
if((arqEntrada = fopen(conta, "r+")) == 0)
{
    //testa se o arquivo pode ser aberto
    printf("Impossível abrir arquivo...\n"); //informa o erro
    exit(1); //finaliza execução
}
arqSaida = fopen(conta, "r+"); //arquivo de saída para salvar no fim do processo.
rewind(arqEntrada); //certifica de que o cursor está no primeiro caracter do arquivo
for(i = 0; !feof(arqEntrada); i++)
{
    //incrementa 'i' enquanto não for fim de arquivo
    memset(textoArquivo, NULO, 100); //inicializa e/ou limpa string 'textoArquivo'
    fgets(textoArquivo, 101, arqEntrada); //pega uma string de 100 caracteres
    if(linha == 1 && linhaArq == 1)
    {
        //se a linha 1 for a escolhida, vai modificar a linha 1, que é número da conta dentro do arquivo
        fprintf(arqSaida, "conta: %.0f\n", linhaValor); //imprime dentro do arquivo, a linha correspondente
        linha++; //incrementa o contador de linhas
        continue; //volta ao início do loop sem executar o resto do código abaixo
    }

    else if(linha == 2 && linhaArq == 2)
    {
        //se a linha 2 for a escolhida, vai modificar a linha 2, que é senha
        fprintf(arqSaida, "senha: %.0f\n", linhaValor); //imprime dentro do arquivo, a linha correspondente
        linha++; //incrementa o contador de linhas
        continue; //volta ao início do loop sem executar o resto do código abaixo
    }

    else if(linha == 3 && linhaArq == 3)
    {
        //se a linha 3 for a escolhida, vai modificar a linha 3
        fprintf(arqSaida, "saldo: %.2f\n", linhaValor); //imprime dentro do arquivo, a linha correspondente
        linha++; //incrementa o contador de linhas
        continue; //volta ao início do loop sem executar o resto do código abaixo
    }

    else if(linha == 4 && linhaArq == 4)
    {
        //se a linha 4 for a escolhida, vai modificar a linha 4
        fprintf(arqSaida, "limite: %.2f\n", linhaValor); //imprime dentro do arquivo, a linha correspondente
        linha++; //incrementa o contador de linhas
        continue; //volta ao início do loop sem executar o resto do código abaixo
    }

    linha++; //incrementa o contador de linhas
    fputs(textoArquivo, arqSaida); //coloca a string 'textoArquivo' no arquivo arqsaida
}
fclose(arqEntrada); //fecha a stream 'arqEntrada'
fclose(arqSaida); //fecha a stream 'arqSaida'
}

msg()
{
    //mensagem do caixa eletrônico, na primeira tela.
    printf ("|-----|\n");
    printf ("|    CAIXA ELETRÔNICO    |    ADS Bank \xA9    |\n"); //\xA9 corresponde ao código hexadecimal da tabela ASCII que é r de registrado.
    printf ("|                                     |\n");
    printf ("|                                DEUS SEJA LOUVADO!                                |\n");
    printf ("|-----|\n");
    printf ("\n");
}

saldoLimite(char conta[])
{
    //função para ler saldo e limite dentro do arquivo.
    float saldoScan, limite;
    limite = 0.00;
    saldoScan = 0.00;
    system("CLS"); //limpa a tela
    msg();
    saldoScan = leituraArquivo(conta, 3); //3 é a linha que se encontra saldo dentro do arquivo.
    printf(" Saldo: R$%.2f || ", saldoScan); //imprime o saldo na tela
    limite = leituraArquivo(conta, 4); //4 é a linha que se encontra limite no arquivo.
    printf(" Limite de Crédito: R$%.2f\n", limite); //imprime o saldo na tela
}

```

```

printf("\n Pressione algo para continuar."); //exibe mensagem na tela
getch();
}
saque(char conta[])
{ //função para fazer a operação saque dentro do arquivo.
    int n;
    float saldo, saldoScan, sacar, limite;
    saldo = 0.00;
    sacar = 0.00;
    limite = 0.00;
    saldoScan = 0.00;
    inicioTrans: //função que volta aqui
    system("CLS"); //limpa a tela
    msg();
    saldoScan = leituraArquivo(conta, 3); //3 é a linha que se encontra saldo dentro do arquivo.
    printf(" Saldo: R$%.2f || ", saldoScan); //imprime saldo na tela.
    limite = leituraArquivo(conta, 4); //4 é a linha que se encontra limite dentro do arquivo.
    printf(" Limite de Crédito: R$%.2f || Limite de saque: R$1000\n", limite); //imprime o limite e limite de saque.
    printf("\n Cédulas disponíveis\n notas de 100, 50, 20 e 10 Reais\n\n Dígite um valor a ser sacado\n 0 - SAIR\n"); //mensagem para o usuário.
    scanf("%f", &sacar); //função pega o que foi digitado pelo usuário.
    n = sacar; //igual a as duas variáveis
    if(n < 0) //verifica se o saque for negativo.
    {
        printf(" Por favor digite um valor positivo!"); //mensagem para o usuário.
        printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
        getch();
        goto inicioTrans; //volta para o início da operação
    }
    if(n == 0) //se n, que é a variável que coleta o que foi digitado for igual a 0 fecha a operação saque
    {
        return 1; //retorna 1
    } else
    if(n > 1000)
    { //função para ver se o que foi digitado de saque for maior que MIL REAIS ele executa o código abaixo
        printf(" Limite máximo de saque: R$1000"); //imprime uma mensagem para o usuário
        printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
        getch(); //função para parar a tela e esperar o usuário teclar algo para poder sair
        goto inicioTrans; //voltar para o início.
    } else
    {
        if(Rcedula(n) == 1)
        { //função para ver se o que foi digitado de saque corresponde as cédulas indisponíveis.
            printf(" Por favor digite um valor correspondente as cédulas disponíveis."); //imprime uma
mensagem para o usuário
            printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
            getch(); //função para parar a tela e esperar o usuário teclar algo para poder sair
            goto inicioTrans; //voltar para o início.
        } else
        {
            if(sacar > 0)
            { //função para ver se o que foi digitado de saque for maior que ZERO ele executa o código.
                if(sacar > (saldoScan + limite))
                {
                    printf(" Saldo indisponível!\n"); //mensagem para o usuário.
                } else if(sacar > saldoScan)
                { //função para verificar se o saque for maior que o saldo disponível, se for ele fara o
cálculo para subtrair do limite o que exceder o saldo disponível
                    cedula(n); //função que verifica quais cédulas o caixa eletrônico vai expelir
                    saldo = saldoScan - sacar; //cálculo para subtrair o saque com o saldo atual e
salvar na variável saldo.
                    gravarArquivo(conta, 3, saldo); // 3 linha saldo, dentro do arquivo
                    printf("\n Valor sacado R$%.2f", sacar); //imprime valor sacado.
                    limite -= (sacar - saldoScan); //cálculo para subtrair o que excedeu do saldo atual
quando for sacar, e salvar na variável limite.
                    gravarArquivo(conta, 4, limite); // 3 linha saldo, dentro do arquivo

```

```

        }else
        {
            cedula(n); //função que verifica quais cédulas o caixa eletrônico vai expelir
            saldo = saldoScan - sacar; //cálculo para subtrair o saldo atual com o valor de saque
e salvar na variável saldo

            gravarArquivo(conta,3,saldo); // 3 linha saldo, dentro do arquivo
            printf("\n Valor sacado R$%.2f",sacar); //imprime o valor sacado
        }
    }
}

printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
getch(); //função para parar a tela e esperar o usuário teclar algo para poder sair
system("CLS"); //limpar tela
}

deposito(char conta[])
{ //função para operação deposito
    float saldoAtual,saldo,ResultsaldoPositivo,ResultsaldoNegativo,saldoNegativo,saldoScan,depositar,
resultLimite,limite;
    saldoAtual = 0.00;
    saldo = 0.00;
    ResultsaldoPositivo = 0.00;
    ResultsaldoNegativo = 0.00;
    saldoNegativo = 0.00;
    saldoScan = 0.00;
    depositar = 0.00;
    resultLimite = 0.00;
    limite = 0.00;
    inicioDep; //função para voltar aqui.
    system("CLS"); //função para limpar a tela
    msg();
    printf(" Valor a ser depositado\n 0 - SAIR\n "); //mensagem para o usuário
    scanf("%f",&depositar);
    if(depositar < 0) //verifica se o saque for negativo
    {
        printf(" Por favor digite um valor positivo!"); //mensagem para o usuário.
        printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
        getch();
        goto inicioDep; //volta para o início da operação
    }
    if(depositar ==0)
    { //se depositar for igual à 0 retorna 1 e para o processo do código deposito.
        return 1; //retorna 1
    }

    saldoScan = leituraArquivo(conta,3); // 3 é a linha que se encontra saldo no arquivo.
    saldoAtual = saldoScan;
    saldoScan +=depositar;
    gravarArquivo(conta,3,saldoScan); // 3 linha saldo
    printf(" Saldo atual:R$%.2f",saldoScan);
    saldoNegativo = saldoScan;
    if(saldoNegativo <0)
    { //se o saldo atual for negativo ele executa.
        ResultsaldoNegativo = saldoNegativo + depositar; //somar o saldo negativo com o depósito
        saldoNegativo = saldoNegativo *-1; //modulo tornar o valor dentro de saldonegativo em um valor positivo.
        resultLimite = saldoNegativo + ResultsaldoNegativo; //salvar na variável resultLimite a soma do
saldoNegativo com ResultadoNegativo
        limite = leituraArquivo(conta,4); //4 é a linha que se encontra limite no arquivo.
        limite +=resultLimite; //soma da variável lime com resultLimite
        gravarArquivo(conta,4,limite); // 4 é a linha limite, e grava dentro do arquivo o valor da variável
limite
    }else
    {
        if(saldoNegativo > 0)
        { //função para verificar se o valor da variável saldoNegativo for positivo
            if(saldoAtual > 0)
            { //função para verificar se o valor da variável saldoAtual for positivo

```

```

        gravarArquivo(conta,3,saldoScan);//gravar dentro do arquivo a variável saldoScan
    }else
    {
        //função que executa se o saldoAtual não for posito
        ResultsaldoPositivo = saldoAtual + depositar;
        resultLimite = depositar - ResultsaldoPositivo;
        limite = leituraArquivo(conta,4);//4 é a linha que se encontra limite no arquivo.
        limite +=resultLimite;//cálculo
        gravarArquivo(conta,4,limite);// 4 é a linha limite
    }
}
}else
if(saldoNegativo == 0)
{
    //função para ver se o saldo é igual a 0.
    resultLimite = saldoAtual * -1;
    limite = leituraArquivo(conta,4);//4 é a linha que se encontra limite no arquivo.
    limite +=resultLimite;//soma as variaveis limite + resultArquivo
    gravarArquivo(conta,4,limite);// 4 é a linha limite
}

printf("\n\n Pressione algo para continuar.");//exibe mensagem na tela
getch();//função para esperar o usuário teclar algo para proceuir com o processo
system("CLS");//função para limpar a tela
}

transferencia(char conta[])
{
    //função para fazer a operação transferência dentro do arquivo
    char nconta[7];
    int sen;
    float valorTransSum,saldoSub,valorTrans,saldoAtual,saldoScan;
    valorTransSum = 0.00;
    saldoSub = 0.00;
    valorTrans = 0.00;
    saldoAtual = 0.00;
    saldoScan = 0.00;
    inicio:
    system("CLS");//função para limpar a tela
    msg();
    saldoScan = leituraArquivo(conta,3);// 3 é a linha que se encontra saldo dentro do arquivo.
    printf(" Conta atual: %s || Saldo: %.2f\n",conta,saldoScan);
    printf("\n Digite o número da conta com 6 dígitos a ser transferido o dinheiro.\n 0 - SAIR\n Conta para
transferir.: ");
    fflush(stdin);//função para limpar a streaming do teclado
    gets(nconta);//função para pegar o que o usuário digitar
    sen = atoi(nconta);//função para converter string para int
    if(sen == 0)
    {
        //sair da operação 4 transferência
        return 1;
    }
    if(strlen(nconta)<6)
    {
        //verifica se a string conta possui 6 digitos.
        printf(" por favor digite 6 números!");//mensagem para o usuário.
        printf("\n\n Pressione algo para continuar.");//exibe mensagem na tela
        getch();
        goto inicio;//volta para o início se a conta não tiver 6 digitos.
    }
    if(verificarCont(nconta) == 2)
    {
        printf("\n Conta não existe! por favor digite um número diferente.");//mensagem para o usuário.
        printf("\n\n Pressione algo para continuar.");//exibe mensagem na tela
        getch();
        goto inicio;//volta ao início se a conta já existir
    }
    inicioVtrans:
    printf(" Digite o valor a ser trasnferido.\n R$ ");//mensagem para o usuário.
    fflush(stdin);//função para limpar a streaming do teclado
    scanf("%f",&valorTrans);
    if(valorTrans < 0)//verifica se o saque for negativo
    {
        printf(" Por favor digite um valor positivo!");//mensagem para o usuário.
    }
}

```

```

printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
getch();
system("cls"); //função para limpar tela
msg();
printf(" Conta atual: %s || Saldo: %.2f\n", conta, saldoScan);
printf("\n Digite número da conta 6 dígitos a ser transferido o dinheiro.\n 0 - SAIR\n"); //mensagem para
o usuário.
printf(" Conta para transferir.: %s\n", nconta);
goto inicioVtrans; //volta para o início da operação
}
if(valorTrans > 0)
{ //se o valor da transferência for maior que 0, executa o código abaixo
    saldoScan = leituraArquivo(conta, 3); // 3 é a linha que se encontra saldo dentro do arquivo.
    saldoAtual = saldoScan; //igualar o valor de uma variável na outra
    if(valorTrans > saldoAtual)
    { //função para verificar se o valor de transferência é maior que o saldo do cliente.
        printf(" Saldo indisponível!\n"); //mensagem para o usuário.
        printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
        getch();
    }
    else
    { //se caso contrário o valor da transferência for menor ou igual o saldo do cliente o código executa
        saldoSub = saldoAtual - valorTrans; //valor do saldo subtraído com o valor da transferência
        gravarArquivo(conta, 3, saldoSub); // 3 é a linha saldo
        valorTransSum = leituraArquivo(nconta, 3); // 3 é a linha que se encontra saldo no arquivo.
        valorTransSum += valorTrans;
        gravarArquivo(nconta, 3, valorTransSum); // 3 é a linha saldo
        printf(" Transferência feita com sucesso!"); //mensagem para o usuário.
        printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
        getch();
    }
}
}
contratarLimite(char conta[])
{ //função para contratar limite dentro do arquivo
    int limiteOPC;
    float limiteScan;
    limiteOPC = 0;
    limiteScan = 0.00;
    do
    {
        system("CLS");
        msg();
        printf(" Escolha um limite a ser contratado.\n 1 - R$500\n 2 - R$1,000\n 3 - R$1,500\n 0 - SAIR\n ");
        //mensagem para o usuário.
        scanf("%i", &limiteOPC); //função para pegar o que o usuário digitar
        switch(limiteOPC)
        { //função para contratar limite com valores pré-definidos
            case 0:
                break;
            case 1:
                limiteScan = 500.00;
                gravarArquivo(conta, 4, limiteScan); // 3 é a linha saldo
                printf(" Limite de Crédito:R$%.2f Contratado.", limiteScan); //mensagem para o usuário.
                printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
                limiteOPC = 0;
                getch();
                break;
            case 2:
                limiteScan = 1000.00;
                gravarArquivo(conta, 4, limiteScan); // 3 é a linha saldo
                printf(" Limite de Crédito:R$%.2f Contratado.", limiteScan); //mensagem para o usuário.
                printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
                limiteOPC = 0;
                getch();
                break;

```

```

    case 3:
        limiteScan = 1500.00;
        gravarArquivo(conta,4,limiteScan); // 3 é a linha saldo
        printf(" Limite de Crédito:R$%.2f Contratado.",limiteScan); //mensagem para o usuário.
        printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
        limiteOPC =0;
        getch();
    break;
    default :
        system("CLS");
        msg();
        printf(" Por favor digite uma operação válida!"); //mensagem para o usuário.
        printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
        getch();
    break;
}
}while(limiteOPC!=0); //função que é um ciclo, só para quando o usuário digitar 0.
system("CLS");
}
cont(char conta[])
{ //função conta, operações correspondentes a ela.
    FILE *arquivo; //ponteiro para apontar onde está o arquivo, dentro da memória ram, para modificá-lo
    int opcao, limiteOPC, saq, dep, trans;
    arquivo = fopen(conta, "r+"); //para verificar se o arquivo existe se existir ele abre.
    if(arquivo==NULL) //verifica se o arquivo é vazio caso for ele cria um arquivo novo. caso contrario ele
    apenas abre.
    {
        arquivo = fopen(conta, "w"); //abrir arquivo com permissão de escrita.
    } else
    {
        arquivo = fopen(conta, "r+"); //abrir arquivo com permissão de leitura e escrita
    }
do
{
    system("CLS");
    msg();
    arquivo = fopen(conta, "r+"); //abre o arquivo com permissão de leitura e escrita
    printf("O que deseja fazer:\n 1 - Consultar Saldo/limite \n 2 - Saque\n 3 - Depósito\n 4 -
Transferência\n 5 - Limite de crédito\n 0 - SAIR\n "); //mensagem para o usuário.
    scanf("%d",&opcao); //ele pega o que foi digitado pelo usuário na tela
    switch(opcao)
    {
        case 1:
            saldoLimite(conta); //executa a função pré-definida que exibe saldo e limite
            break;
        case 2:
            saq = saque(conta); //abre a função saque e também a variável saq recebe o valor retornado pela
função
            if(saq == 1)
            {
                break; //para o ciclo
            }
            break; //parar operação 2 que é saque
        case 3:
            dep = deposito(conta); //abre a função deposito e também a variável depósito recebe o valor
retornado pela função
            if(dep == 1)
            {
                break; //para o ciclo
            }
            break;
        case 4:
            trans = transferencia(conta); //abre a função transferência e também a variável trans recebe o
valor retornado pela função
            if(trans == 1)
            {

```

```

        break;//para o ciclo
    }
    break;
case 5:
    contratarLimite(conta);//abre a função contratarLimite
    break;
case 0:
    fclose(arquivo);//fecha o arquivo para não dar problemas futuros
    return 0;
    break;
default :
    system("CLS");
    printf("Por favor digite uma operação válida");//mensagem para o usuário
    printf("\n\n Pressione algo para continuar.");//exibe mensagem na tela
    getch();
    break;
}
}while(opcao!=0);
}
cedula(int n)
{
//função para contar cédulas, e exibi-las
    int n100, n50, n20, n10;
    n100 = n/100;
    n50 = (n%100)/50;
    n20 = ((n%100)%50)/20;
    n10 = (((n%100)%50)%20)/10;
    if(n100 > 0)
    {
//função para verificar quantas cédulas o caixa eletrônico var expelir, as funções de baixo faz a mesma
        coisa.
        printf(" cédulas(s) de R$100 : %i\n",n100);//mensagem para o usuário
    }
    if(n50 > 0)
    {
        printf(" cédulas(s) de R$50 : %i\n",n50);//mensagem para o usuário
    }
    if(n20 > 0)
    {
        printf(" cédulas(s) de R$20 : %i\n",n20);//mensagem para o usuário
    }
    if(n10 > 0)
    {
        printf(" cédulas(s) de R$10 : %i\n",n10);//mensagem para o usuário
    }
}
Rcedula(int n)
{
//função que verifica se o usuário for sacar um valor diferente das cédulas disponiveis 100,50,20 e 10 reais
    int n5, n4, n3, n2,n1;
    n5 = (((((n%100)%50)%20)%10)/5);
    n4 = (((((n%100)%50)%20)%10)%5)/4;
    n3 = ((((((n%100)%50)%20)%10)%5)%4)/3;
    n2 = (((((((n%100)%50)%20)%10)%5)%4)%3)/2;
    n1 = (((((((((n%100)%50)%20)%10)%5)%4)%3)%2)/1;
    if(n5 > 0)
    {
//função para verificar se o usuário escolheu um valor que não tem pode ser sacado porque não há
        cédulas no caixa eletrônico, mesma coisa nos código abaixo
        return 1;
    }else
    if(n4 > 0)
    {
        return 1;
    }else
    if(n3 > 0)
    {
        return 1;
    }else
    if(n2 > 0)

```



```

        {
            return 1;
        }else
            if(n1 > 0)
            {
                return 1;
            }
    }
senha(char conta[])
{
    //função para verificar a senha digitada pelo usuário, para poder acessar a conta
    char c,login[10], senha[10],lgn[15],shna[15];
    int x=4, a=1, b=1,i,k,shnal;
    FILE *arq;
    arq = fopen(conta,"r");
    if (arq == NULL) // Se houve erro na abertura
    {
        printf(" Conta não existe\n");//mensagem para o usuário.
        getch();
        return 0;
    }
    shnal = leituraArquivo(conta,2);//converter int para char
    itoa(shnal,shna,10);//converter int para char
    for(k=0;k<4;k++)
    {
        inil:
        fflush(stdin);//limpa a streaming do teclado
        printf(" Senha: ");
        i=0;
        do{
            c=getch();
            if(isprint(c))
            {
                //Analisa se o valor de c e imprimível
                senha[i]=c;
                i++;
                printf("*");//Imprimindo apenas o asterisco *
            }
            else if(c==8&& i)
            {
                //8 e o caractere BackSpace na tabela ASCII, && a analisa se i e diferente de 0
                senha[i]='\0';
                i--;
                printf("\b \b");//Apagando os caracteres digitados
            }
        }while(c!=13);//13 e o valor de ENTER na tabela ASCII
        senha[i]='\0';
        if(strlen(senha)<6)
        {
            //verifica se a string conta possui 6 dígitos.
            printf("\n por favor digite 6 números!");
            getch();
            system("CLS");
            msg();
            printf(" Por favor digite o número da conta.\n 0 - Sair\n Conta: %s\n",conta);
            goto inil;//volta para o inil, se a senha não possuir 6 dígitos.
        }

        b = strcmp(senha,shna);//compara se a avariável string login é igual a variável string l. e se for iguais
        retorna 0
        if(b==0)
        {
            printf("\n          LOADING...");
            cont(conta);
            break;
        }else
        {
            system("CLS");
            msg();
            x--;
            printf("\a tente novamente chance(s) %i\n",x);
        }
    }
}

```

```

printf(" Conta: %s\n",conta);
if(x == 0)
{
    printf("\n          CONTA BLOQUEADA!");
    printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
    getch();
}
}
}
}

createCont(char conta[], char senha[])
{ //função para criar uma conta nova
    FILE *arq;
    arq = fopen(conta,"w");
    int senha,contal;
    contal = atoi(conta); //converter string para int
    gravarArquivo(conta,1,contal); //linha 1 é a conta dentro do arquivo
    senha = atoi(senha); //converter string para int
    gravarArquivo(conta,2,senha); //linha 2 é a senha dentro do arquivo
    gravarArquivo(conta,3,0); //linha é a 3 saldo
    gravarArquivo(conta,4,0); //linha é a 4 limite
}

verificarCont(char conta[])
{ //função para verificar a conta, se retornar 1 a conta existe, caso contrário não existe
    FILE *arq;
    arq = fopen(conta,"r");
    if(arq == NULL)
    {
        return 2;
    }else
    {
        return 1;
    }
}

/*****EASTER EGG*****/

creditos()
{ //função para exibir credits
printf("          v                                \n");
printf("          uBRBRB                                \n");
printf("          RBRBRBRr                                \n");
printf("          rBRBRBRBRB                                \n");
printf("          RBRBRBRBRB                                \n");
printf("          RBRBRBRBF                                \n");
printf("          J7DRBRBRBJ                                \n");
printf("          1BRBRBRBRB.  RBKBR3                                \n");
printf("          RBRBRBRBRBRBRB7  SZ2RE                                rB \n");
printf("          HBRBRBRBRBRBRBRBRE DR sBO                                JRB \n");
printf("          RBRBRBRBRBRBRBRBRBRUBR: BR,                                BRBR2 \n");
printf("          PBRBRBRBRBRBRBRBRBRBRBM ORB                                BRBRBR \n");
printf("          KBRBRBRBRBRBRBRBRBRBRB RB  E RBRZ                                \n");
printf("          GBRBRBRBRBRBRBRBRBRBRB RBKBRBRB                                \n");
printf("          HBRBRBRBRBRBRBRBRBRBRB rBRBRBR                                \n");
printf("          .RBRBRBRBRBRBRBRBRBRBRBRBRORBRi ;uBRBRBRBR                                \n");
printf("          RBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRB \n");
printf("          BRBRBRBRBRBRBRB ;RBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBr \n");
printf("          RBRBRBRBRBRBRB1 RBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRB \n");
printf("          JRBRBRBRBRBRB1BRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRJ \n");
printf("          RBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRB \n");
printf("          RBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRB \n");
printf("          RBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRBRB7RB\n");
printf("          RBRBRBRBRBRBRBRBRBRBRBRBRBRBJ MRBRBRBP BRBRBRBRBGi ,SBRBW\n");
printf("          RBRBRBRBRBRBRBRBRBRBRBRBRRL .BRBRBRB xRBLPRBRBRBRBRB \n");
printf("          RBRBRBRBRBRBRBRBRBRBRB RBRBRBR2 RBRBRBR \n");
printf("          URJ BRBRBRBR RBRB \n");
printf("          RB B RB xBRBRBRB XH \n");
printf("          BRBR LRL BRBRK RBRBRBRB \n");

```

```
printf("                BR77BRBRBRBRBR0 BR          BRBRBRBR \n");
printf("                u 7RBRBRBRBRBL           c       LBRBR \n");
printf("                :RBRBRBRBRBRBRBRBRBRBRBRBRBRBR RBRD \n");
printf("                RBRBRBDR0DL      DR        73GSP201XRB.   BRBR \n");
printf("                BR              RB             RRB XRBRBR \n");
printf("                ;               BR            RBRBRBRB \n");
printf("                DB              SRBRBRBRBRB \n");
printf("                BR              .RRBRBR. \n");
printf("                :B \n");
printf("/////////////////////////////////////////\n");
printf("*\n");
printf("# ##### # ##### # # # # # \n");
printf("# # # # # # # # # # # # # # \n");
printf("# # # # # # # # # # # # # # \n");
printf("# # # # # ##### # # # # # # # # \n");
printf("# ##### # # # # # ##### # # # # \n");
printf("# # # # # # # # # # # # # # \n");
printf("# # # # # # # # # # # # # # \n");
printf("# # ##### # # # # # # # # \n");
printf("*\n");
printf("By Wesley Hjpr *\n");
printf("/////////////////////////////////////////\n");
printf("\n");
}

/*****EASTER EGG*****/

cadastro()
{ //função para fazer o cadastro do cliente
    int opcao,i,cont;
    char nconta[7],nsenha[7],c,d,s=1,senha[7];
    inicio:
    system("CLS");
    msg();
    printf(" Por favor digite o número da conta com 6 dígitos.\n 0 - Sair\n Conta: "); //mensagem para o usuário.
    fflush(stdin); //função para limpar a streaming do teclado.
    gets(nconta); //função para pegar o que o usuário digitar
    cont = atoi(nconta);
    if(cont == 0)
    { //sair
        return 1;
    }
    if(strlen(nconta)<6)
    { //verifica se conta possui 6 dígitos.
        printf(" por favor digite 6 números!"); //mensagem para o usuário.
        printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
        getch();
        goto inicio; //volta para o inicio se a conta nao tiver 6 digitos.
    }
    if(verificarCont(nconta) == 1)
    {
        printf(" Conta já existe! por favor digite um número diferente."); //mensagem para o usuário.
        printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
        getch();
        goto inicio; //volta ao início se a conta já existir
    }
    iniciol:
    printf(" Por favor digite a senha da conta com 6 dígitos. \n Senha: "); //mensagem para o usuário.
    fflush(stdin); //limpar streaming do teclado
    i=0;
    do
    {
        c=getch();
        if(isprint(c))
        { //Analisa se o valor de c e imprimível
            nsenha[i]=c;
            i++;
            printf("*"); //Imprimindo apenas o asterisco *
        }
    }
```

```

        else if(c==8&& i)
        {
            //8 e o caractere BackSpace na tabela ASCII, && a analisa se i e diferente de 0
            nsenha[i]='\0';
            i--;
            printf("\b \b");//Apagando os caracteres digitados
        }
    }while(c!=13);//13 e o valor de ENTER na tabela ASCII
    nsenha[i]='\0';
    if(strlen(nsenha)<6)
    {
        //verifica se conta possui 6 dígitos.
        printf("\n por favor digite 6 números!");//mensagem para o usuário.
        printf("\n\n Pressione algo para continuar.");//exibe mensagem na tela
        getch();
        system("CLS");
        msg();
        printf(" Por favor digite o número da conta com 6 dígitos.\n 0 - Sair\n Conta: %s\n",nconta);//mensagem
para o usuário.
        goto inicio1;//volta para o inicio1 se a senha não tiver 6 dígitos.
    }
    printf("\n digite a senha novamente.\n Senha: ");//mensagem para o usuário.
    i=0;
    do
    {
        d=getch();
        if(isprint(d))
        {
            //Analisa se o valor de c e imprimível
            senhal[i]=d;
            i++;//incrementa 1 a variável
            printf("*");//Imprimindo apenas o asterisco * quando o usuário digitar
        }
        else if(d==8&& i)
        {
            //8 e o caractere BackSpace na tabela ASCII, && a analisa se i e diferente de 0
            senhal[i]='\0';
            i--;//decrementa 1 a variável
            printf("\b \b");//Apagando os caracteres digitados
        }
    }while(d!=13);//13 e o valor de ENTER na tabela ASCII
    senhal[i]='\0';
    s = strcmp(senhal,nsenha);//compara se a avariável string login é igual a variável string l. e se for iguais
retorna 0
    if(s==0)
    {
        createCont(nconta,nsenha);
        printf("\n Conta feita com sucesso!");//mensagem para o usuário.
        printf("\n\n Pressione algo para continuar.");//exibe mensagem na tela
        getch();
    }else
    {
        printf("\n senha não corresponde! digite novamente ");//mensagem para o usuário.
        printf("\n\n Pressione algo para continuar.");//exibe mensagem na tela
        getch();
        system("CLS");
        msg();
        printf(" Por favor digite o número da conta com 6 dígitos.\n 0 - Sair\n Conta: %s\n",nconta);
//mensagem para o usuário.
        goto inicio1;
    }
}

entrarConta()
{
    //função para entrar na conta
    int cont;
    char nconta[7];
    ini:
    system("CLS");
    msg();//função para exibir a mensagem do caixa eletrônico
    printf(" Por favor digite o número da conta.\n 0 - Sair\n Conta: ");//mensagem para o usuário.

```

```

fflush(stdin); //função para limpar a streaming do teclado
gets(nconta); //função para pegar o que o usuário digitar
cont = atoi(nconta); //função para converter string em int
if(cont == 0)
{ //sair
    return 1; // retorna 1
}
if(strlen(nconta) < 6)
{ //verifica se conta possui 6 dígitos.
    printf(" por favor digite 6 números!"); //mensagem para o usuário.
    printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
    getch();
    goto ini; //volta para o ini se a conta não tiver 6 dígitos.
}
senha(nconta); //executa função para verificar se o usuário digitou a senha corretamente
}
int main()
{ //função principal do programa
    int opcao, i, cont, cdtro, entrcnta;
    char nconta[7], nsenha[7], c, d, s=1, senhal[7];
    setlocale(LC_ALL, "portuguese"); //função para poder funcionar a acentuação no software.
do
{ //função que faz um ciclo
    system("CLS"); //função para limpar a tela
    msg(); //função para exibir mensagem do caixa eletrônico
    printf(" O que deseja fazer:\n 1 - Cadastrar Conta.\n 2 - Entrar Conta.\n 0 - Sair\n "); //mensagem para
o usuário.
    scanf("%i", &opcao); //função para pegar o que o usuário digitar
    switch(opcao)
    { //função para verificar a opção digitada pelo usuário
        case 0:
            break; //função para parar o ciclo
        case 1:
            cdtro = cadastro(); //abre a função cadastro e também a variável cdtro recebe o valor retornado
pela função
            if(cdtro == 1)
            { //se cdtro retornar 1, vai parar o caso 1 das operações
                break; //função para parar o ciclo
            }
            break; //função para parar o ciclo
        case 2:
            entrcnta = entrarConta(); //abre a função transferência e também a variável entrcnta recebe o
valor retornado pela função
            if(entrcnta == 1)
            { //se entrcnta retornar 1 para o caso 2
                break; //função para parar o ciclo
            }
            break; //função para parar o ciclo
    }
    /*****EASTER EGG*****/
    case 1337:
        system("CLS"); //função para limpar a tela
        credits(); //função para exibir os credits
        printf(" Pressione algo para continuar."); //exibe mensagem na tela
        getch();
        break; //função para parar o ciclo
    /*****EASTER EGG*****/
    default :
        system("CLS"); //função para limpar a tela
        printf(" Por favor digite uma operação válida!"); //mensagem para o usuário.
        printf("\n\n Pressione algo para continuar."); //exibe mensagem na tela
        getch();
        break; //função para parar o ciclo
    }
}while(opcao != 0); //termina o ciclo quando o usuário digitar 0
return 0; //retorna 0 para finalizar o programa
}

```