

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

struct info_livro{
    char sigla;
    int reg;
};

typedef struct aluno{
    char nome[80];
    char RA[7];
    int emprestado;
    int reservado;
    struct info_livro tabela[4];
}aluno;

struct info_aluno{
    char sigla;
    char RA[7];
    int dia_ret;
    int mes_ret;
    int dia_dev;
    int mes_dev;
};

typedef struct livro{
    int reg_livro;
    char titulo[80];
    char autor[80];
    struct info_aluno status[2];
}livro;

int verificativ();
int verificaalu();
void alocaliv(livro **p, int tam);
void alocaalu(aluno **p, int tam);
void cadastra_livro(livro *p, int cont);
void cadastra_aluno(aluno *pa, livro *p, int cont);
void gravaliv(livro *p);
void gravaalu(aluno *p);
void mostra_aluno_Total(aluno *p, int cont);
void mostra_aluno_Parcial(aluno *p, int cont);

```

```
void mostra_livro_Total(livro *p, int cont);
void mostra_livro_Parcial_Status(livro *p, int cont);
void mostra_livro_Parcial_Titulo(livro *p, int cont);
void tratamento(char um[], char dois[]);
```

```
int main(void)
{
    int op, contliv=0,contalu=0;
    livro *liv=NULL;
    alocaliv(&liv,1);
    aluno *alu=NULL;
    alocaalu(&alu,1);
    contliv=verificaliv();
    contalu=verificaalu();
    do{
        system("cls");
        printf("\n[1]Cadastro Aluno\n[2]Mostra Aluno Total\n[3]Mostra Aluno Parcial\n[4]Cadastro
Livro\n[5]Mostra Livro Total\n[6]Mostra Livro Status\n[7]Mostra Livro Titulo\n[8]Sair\n:");
        scanf("%i", &op);
        fflush(stdin);
        switch(op){
            case 1:
                if(contliv==0)
                {
                    system("cls");
                    printf("\nPrimeiro cadastre um livro.....\n\n\n");
                    system("pause");
                }
                else
                {
                    cadastra_aluno(alu, liv, contliv);
                    contalu++;
                }
                break;
            case 2:
                mostra_aluno_Total(alu, contalu);
                break;
            case 3:
                mostra_aluno_Parcial(alu, contalu);
                break;
            case 4:
                cadastra_livro(liv,contliv+1);
                contliv++;
        }
    }
}
```

```

        break;
    case 5:
        mostra_livro_Total(liv,contliv);
        break;
    case 6:
        mostra_livro_Parcial_Status(liv,contliv);
        break;
    case 7:
        mostra_livro_Parcial_Titulo(liv,contliv);
        break;
    }
}while(op!=8);
}

```

```

void alocaliv(livro **p, int tam){

    if((*p=(livro*)realloc(*p,tam*sizeof(livro)))==NULL)
    {
        system("cls");
        printf("\nErro de Alocação.....\n\n\n");
        system("pause");
        exit(1);
    }
}

```

```

void alocaalu(aluno **p, int tam){

    if((*p=(aluno*)realloc(*p,tam*sizeof(aluno)))==NULL)
    {
        system("cls");
        printf("\nErro de Alocação.....\n\n\n");
        system("pause");
        exit(1);
    }
}

```

```

verificaliv(){
    FILE *fptr=NULL;
    long int cont=0;
    if((fptr=fopen("Etapa2.bin","rb"))==NULL)
        return cont;
    else
    {

```

```

        fseek(fp,0,2);
        cont=ftell(fp)/sizeof(livro);
        return cont;
    }
}

```

```

verificaalu(){
    FILE *fp=NULL;
    long int cont=0;
    if((fp=fopen("Biblioteca.bin","rb"))==NULL)
        return cont;
    else
    {
        fseek(fp,0,2);
        cont=ftell(fp)/sizeof(aluno);
        return cont;
    }
}

```

```

gravaliv(livro *p){
    FILE *fp=NULL;

    if((fp=fopen("Etapa2.bin","ab"))==NULL)
    {
        system("cls");
        printf("\n Erro na abertura do arquivo.....\n\n\n");
        system("pause");
    }
    else
    {
        fwrite(p,sizeof(livro),1,fp);
        system("cls");
        printf("\nCadastro efetuado com sucesso.....\n\n\n");
        system("pause");
    }
    fclose(fp);
}

```

```

gravaalu(aluno *p){
    FILE *fp=NULL;

    if((fp=fopen("Biblioteca.bin","ab"))==NULL)
    {

```

```

        system("cls");
        printf("\n Erro na abertura do arquivo.....\n\n\n");
        system("pause");
    }
    else
    {
        fwrite(p,sizeof(aluno),1,fptr);
        printf("\nCadastro efetuado com sucesso.....\n\n\n");
        system("pause");
    }
    fclose(fptr);
}

```

```

void cadastra_livro(livro *p, int cont){
    int x=0;
    p->reg_livro=cont;
    system("cls");
    printf("Registro do livro: %i", p->reg_livro);
    printf("\nTitulo: ");
    fflush(stdin);
    gets(p->titulo);
    printf("\nAutor:");
    gets(p->autor);
    fflush(stdin);
    for(x=0;x<2;x++)
        (p->status+x)->sigla='L';
    gravaliv(p);
}

```

```

void cadastra_aluno(aluno *pa, livro *p, int cont){
    char aux[80], aux2, aux3[7], aux4[80], aux5[80], op;
    int i=0, z=0, y=0, x=0, j=0, f=0, q=0, l=0, h=0;
    pa->emprestado=0;
    pa->reservado=0;
    for(h=0;h<4;h++)
    {
        (pa->tabela+h)->sigla='L';
        (pa->tabela+h)->reg=0;
    }
    system("cls");
    printf("\n Nome do aluno:");
    gets(pa->nome);
    fflush(stdin);
}

```

```

printf("\n RA: ");
gets(pa->RA);
fflush(stdin);
do{
    system("cls");
    printf("\nTitulo ou Autor do livro desejado: ");
    gets(aux);
    fflush(stdin);
    tratamento(aux,aux);
    (pa->tabela+l)->sigla=toupper((pa->tabela+l)->sigla);
    FILE *fptr=NULL;
    if((fptr=fopen("Etapa2.bin","rb+"))==NULL)
    {
        system("cls");
        printf("\n Erro na abertura do arquivo.....\n\n\n");
        system("pause");
    }
    else
    {
        for(i=0;i<cont;i++)
        {
            fseek(fptr,i*sizeof(livro),0);
            fread(p,sizeof(livro),1,fptr);
            tratamento(p->autor,aux4);
            tratamento(p->titulo,aux5);
            if(((strcmp(aux,aux4))==0)||((strcmp(aux,aux5))==0))
            {
                if((p->status+0)->sigla=='L')
                {
                    j=0;
                    (pa->tabela+l)->sigla='E';
                    system("cls");
                    printf("\n Livro encontrado e livre.....\n\n\n");
                    system("pause");
                    strcpy((p->status+j)->RA,pa->RA);
                    do{
                        system("cls");
                        printf("\n Mes de retirada: ");
                        scanf("%i", &((p->status+j)->mes_ret));
                        fflush(stdin);
                        if(((p->status+j)->mes_ret)<1 || ((p->status+j)->mes_ret)>12)
                        {
                            system("cls");

```

```

        printf("\n Digite um mes valido.....\n\n\n");
        system("pause");
    }
}while(((p->status+j)->mes_ret)<1 || ((p->status+j)->mes_ret)>12);

if(((p->status+j)->mes_ret)==2||((p->status+j)->mes_ret)==4||((p->status+j)->mes_ret)==6||((p->
status+j)->mes_ret)==9||((p->status+j)->mes_ret)==11)
{
    if(((p->status+j)->mes_ret)==2)
    {
        do{
            printf("\n Dia de retirada: ");
            scanf("%i", &((p->status+j)->dia_ret));
            fflush(stdin);

            ((p->status+j)->dia_dev)=((p->status+j)->dia_ret)+7;

            if(((p->status+j)->dia_dev)>28)
            {
                ((p->status+j)->dia_dev)-=28;
                ((p->status+j)->mes_dev)=((p->status+j)->mes_ret)+1;
                if(((p->status+j)->mes_dev)>12)
                {
                    ((p->status+j)->mes_dev)-=12;
                }
            }
            else
            {
                ((p->status+j)->mes_dev)=((p->status+j)->mes_ret);
            }
            if(((p->status+j)->dia_ret)<=28 && ((p->status+j)->dia_ret)>=1)
            {
                system("cls");
                printf("\n Devolucao: Dia %i - Mes %i \n\n\n",
(p->status+j)->dia_dev, (p->status+j)->mes_dev);
                (p->status+j)->sigla=(pa->tabela+l)->sigla;
                system("pause");
                fseek(fptra,i*sizeof(livro),0);
                fwrite(p,sizeof(livro),1,fptra);
                f+=1;
                i=100;
            }
            else

```

```

        {
            system("cls");
            printf("\n Erro, fevereiro tem apenas 28 dias.....\n\n\n");
            system("pause");
        }
    }while(((p->status+j)->dia_ret)>28 || ((p->status+j)->dia_ret)<1);
}
else
{
    do{
        printf("\n Dia de retirada: ");
        scanf("%i", &((p->status+j)->dia_ret));
        fflush(stdin);

        ((p->status+j)->dia_dev)=((p->status+j)->dia_ret)+7;

        if(((p->status+j)->dia_dev)>30)
        {
            ((p->status+j)->dia_dev)-=30;
            ((p->status+j)->mes_dev)=((p->status+j)->mes_ret)+1;
            if(((p->status+j)->mes_dev)>12)
            {
                ((p->status+j)->mes_dev)-=12;
            }
        }
        else
        {
            ((p->status+j)->mes_dev)=((p->status+j)->mes_ret);
        }
        if(((p->status+j)->dia_ret)<=30 && ((p->status+j)->dia_ret)>=1)
        {
            system("cls");
            printf("\n Devolucao: Dia %i - Mes %i \n\n\n",
(p->status+j)->dia_dev, (p->status+j)->mes_dev);
            (p->status+j)->sigla=(pa->tabela+l)->sigla;
            system("pause");
            fseek(fptr,i*sizeof(livro),0);
            fwrite(p,sizeof(livro),1,fptr);
            f+=1;
            i=100;
        }
        else
        {

```



```

        system("cls");
        printf("\n Erro, digite um dia valido.....\n\n\n");
        system("pause");
    }
}while(((p->status+j)->dia_ret)>30 || ((p->status+j)->dia_ret)<1);
}
}
else
{
do{
    printf("\n Dia de retirada: ");
    scanf("%i", &((p->status+j)->dia_ret));
    fflush(stdin);

    ((p->status+j)->dia_dev)=((p->status+j)->dia_ret)+7;

    if(((p->status+j)->dia_dev)>31)
    {
        ((p->status+j)->dia_dev)-=31;
        ((p->status+j)->mes_dev)=((p->status+j)->mes_ret)+1;
        if(((p->status+j)->mes_dev)>12)
        {
            ((p->status+j)->mes_dev)-=12;
        }
    }
    else
    {
        ((p->status+j)->mes_dev)=((p->status+j)->mes_ret);
    }
    if(((p->status+j)->dia_ret)<=31 && ((p->status+j)->dia_ret)>=1)
    {
        system("cls");
        printf("\n Devolucao: Dia %i - Mes %i \n\n\n",
(p->status+j)->dia_dev, (p->status+j)->mes_dev);
        (p->status+j)->sigla=(pa->tabela+l)->sigla;
        system("pause");
        fseek(fptr,i*sizeof(livro),0);
        fwrite(p,sizeof(livro),1,fptr);
        f+=1;
        i=100;
    }
    else
    {

```

```

        system("cls");
        printf("\n Erro, digite um dia valido.....\n\n\n");
        system("pause");
    }
    }while(((p->status+j)->dia_ret)>31 || ((p->status+j)->dia_ret)<1);
}
pa->emprestado+=1;
(pa->tabela+l)->reg=p->reg_livro;
l+=1;
}
else
{
    if((p->status+1)->sigla=='L'&&(pa->reservado)<1)
    {
        j=1;
        system("cls");
        printf("\n Livro ja emprestado, gostaria de reservar? <S/N>");
        scanf("%c", &op);
        fflush(stdin);
        if(op=='S'||op=='s')
        {
            (pa->tabela+l)->sigla='R';
            system("cls");
            printf("\n Livro encontrado e livre.....\n\n\n");
            system("pause");
            strcpy((p->status+j)->RA,pa->RA);

if(((p->status+0)->mes_dev)==2||((p->status+0)->mes_dev)==4||((p->status+0)->mes_dev)==6||
((p->status+0)->mes_dev)==9||((p->status+0)->mes_dev)==11)
        {
            if(((p->status+0)->mes_dev)==2)
            {
                ((p->status+j)->dia_ret)=((p->status+0)->dia_dev)+1;

                if(((p->status+j)->dia_ret)>28)
                {
                    ((p->status+j)->dia_ret)-=28;
                    ((p->status+j)->mes_ret)=((p->status+0)->mes_dev)+1;
                    if(((p->status+j)->mes_ret)>12)
                    {
                        ((p->status+j)->mes_ret)-=12;
                    }
                }
            }
        }
    }
}

```

```

else
{
    ((p->status+j)->mes_ret)=((p->status+0)->mes_dev);
}

((p->status+j)->dia_dev)=((p->status+j)->dia_ret)+7;

if(((p->status+j)->dia_dev)>28)
{
    ((p->status+j)->dia_dev)-=28;
    ((p->status+j)->mes_dev)=((p->status+j)->mes_ret)+1;
    if(((p->status+j)->mes_dev)>12)
    {
        ((p->status+j)->mes_dev)-=12;
    }
}
else
{
    ((p->status+j)->mes_dev)=((p->status+j)->mes_ret);
}

if(((p->status+j)->dia_ret)<=28 && ((p->status+j)->dia_ret)>=1)
{
    system("cls");
    printf("\n Reserva efetuada de %i/%i ate %i/%i \n\n\n\n",
(p->status+j)->dia_ret, (p->status+j)->mes_ret, (p->status+j)->dia_dev, (p->status+j)->mes_dev);
    (p->status+j)->sigla=(pa->tabela+l)->sigla;
    system("pause");
    fseek(fptr,i*sizeof(livro),0);
    fwrite(p,sizeof(livro),1,fptr);
    f+=1;
    i=100;
}
}
else
{
    ((p->status+j)->dia_ret)=((p->status+0)->dia_dev)+1;

    if(((p->status+j)->dia_ret)>30)
    {
        ((p->status+j)->dia_ret)-=30;
        ((p->status+j)->mes_ret)=((p->status+0)->mes_dev)+1;
        if(((p->status+j)->mes_ret)>12)

```

```

        {
            ((p->status+j)->mes_ret)-=12;
        }
    }
    else
    {
        ((p->status+j)->mes_ret)=((p->status+0)->mes_dev);
    }

    ((p->status+j)->dia_dev)=((p->status+j)->dia_ret)+7;

    if(((p->status+j)->dia_dev)>30)
    {
        ((p->status+j)->dia_dev)-=30;
        ((p->status+j)->mes_dev)=((p->status+j)->mes_ret)+1;
        if(((p->status+j)->mes_dev)>12)
        {
            ((p->status+j)->mes_dev)-=12;
        }
    }
    else
    {
        ((p->status+j)->mes_dev)=((p->status+j)->mes_ret);
    }

    if(((p->status+j)->dia_ret)<=30 && ((p->status+j)->dia_ret)>=1)
    {
        system("cls");
        printf("\n Reserva efetuada de %i/%i ate %i/%i \n\n\n\n",
(p->status+j)->dia_ret, (p->status+j)->mes_ret, (p->status+j)->dia_dev, (p->status+j)->mes_dev);
        (p->status+j)->sigla=(pa->tabela+l)->sigla;
        system("pause");
        fseek(fptr,i*sizeof(livro),0);
        fwrite(p,sizeof(livro),1,fptr);
        f+=1;
        i=100;
    }
}
}
else
{
    ((p->status+j)->dia_ret)=((p->status+0)->dia_dev)+1;

```

```

if(((p->status+j)->dia_ret)>31)
{
    ((p->status+j)->dia_ret)-=31;
    ((p->status+j)->mes_ret)=((p->status+0)->mes_dev)+1;
    if(((p->status+j)->mes_ret)>12)
    {
        ((p->status+j)->mes_ret)-=12;
    }
}
else
{
    ((p->status+j)->mes_ret)=((p->status+0)->mes_dev);
}

((p->status+j)->dia_dev)=((p->status+j)->dia_ret)+7;

if(((p->status+j)->dia_dev)>31)
{
    ((p->status+j)->dia_dev)-=31;
    ((p->status+j)->mes_dev)=((p->status+j)->mes_ret)+1;
    if(((p->status+j)->mes_dev)>12)
    {
        ((p->status+j)->mes_dev)-=12;
    }
}
else
{
    ((p->status+j)->mes_dev)=((p->status+j)->mes_ret);
}

if(((p->status+j)->dia_ret)<=31 && ((p->status+j)->dia_ret)>=1)
{
    system("cls");
    printf("\n Reserva efetuada de %i/%i ate %i/%i \n\n\n",
(p->status+j)->dia_ret, (p->status+j)->mes_ret, (p->status+j)->dia_dev, (p->status+j)->mes_dev);
    (p->status+j)->sigla=(pa->tabela+l)->sigla;
    system("pause");
    fseek(fptr,i*sizeof(livro),0);
    fwrite(p,sizeof(livro),1,fptr);
    f+=1;
    i=100;
}
}

```

```

        pa->reservado+=1;
        (pa->tabela+l)->reg=p->reg_livro;
        l+=1;
    }
}
else
{
    system("cls");
    printf("\n Livro não pode ser emprestado nem reservado.....\n\n\n");
    system("pause");
}
}
}
}
fclose(fptr);
}
if(i<100)
{
    system("cls");
    printf("\n Erro, livro sem espaço livre ou não encontrado.....\n\n\n");
    system("pause");
}
op='N';
if(f<cont&&(pa->emprestado<3||pa->reservado<1))
{
    system("cls");
    printf("\n Gostaria de Reservar/Empréstimo outro livro?\n <S/N>:");
    scanf("%c", &op);
    fflush(stdin);
}
}while(op=='S'||op=='s');
if(f>0)
    gravaalu(pa);
}

```

```

void mostra_aluno_Total(aluno *p, int cont){

    int i=0, z=0;
    FILE *fptr=NULL;

    if((fptr=fopen("Biblioteca.bin", "rb"))==NULL)
    {

```

```

        system("cls");
        printf("\nErro ao abrir o arquivo.....\n\n\n");
        system("pause");
    }
    else
    {
        system("cls");
        for(i=0;i<cont;i++)
        {
            fseek(fptr,i*sizeof(aluno),0);
            fread(p,sizeof(aluno),1,fptr);
            printf("\n||| Nome - %s | RA - %s | Emprestados - %i | Reservados - %i |||",
(p->nome), p->RA, p->emprestado, p->reservado);
            for(z=0;z<4;z++)
            {
                if((p->tabela+z)->sigla=='E'||(p->tabela+z)->sigla=='R')
                    printf("\n||| Livro %i - reg: %i - sigla: %c |||", z+1, (p->tabela+z)->reg,
(p->tabela+z)->sigla);
                else
                    printf("\n||| Livro %i -----espaco livre---- |||", z+1);
            }
            printf("\n\n\n");
        }
        fclose(fptr);
        printf("\n\n\n\n");
        system("pause");
    }
}

```

```

void mostra_aluno_Parcial(aluno *p, int cont){

```

```

    char aux[7];
    int i=0, z=0;
    system("cls");
    printf("\n RA do aluno: ");
    gets(aux);

```

```

    FILE *fptr=NULL;

```

```

    if((fptr=fopen("Biblioteca.bin", "rb"))==NULL)
    {
        system("cls");
        printf("\nErro ao abrir o arquivo.....\n\n\n");
    }

```

```

        system("pause");
    }
    else
    {
        system("cls");
        for(i=0;i<cont;i++)
        {
            fseek(fptr,i*sizeof(aluno),0);
            fread(p,sizeof(aluno),1,fptr);
            if((strcmp(aux, p->RA))==0)
            {
                printf("\n||| Nome - %s | RA - %s | Emprestados - %i | Reservados - %i |||",
(p->nome), p->RA, p->emprestado, p->reservado);
                for(z=0;z<4;z++)
                {
                    if((p->tabela+z)->sigla=='E'||(p->tabela+z)->sigla=='R')
                        printf("\n||| Livro %i - reg: %i - sigla: %c |||", z+1, (p->tabela+z)->reg,
(p->tabela+z)->sigla);
                    else
                        printf("\n||| Livro %i -----espaco livre---- |||", z+1);

                }
                printf("\n\n\n");
                i=100;
            }
        }
        fclose(fptr);
        if(i<100)
            printf("\n RA nao encontrado.....\n\n\n");
        system("pause");
    }
}

```

```

void mostra_livro_Total(livro *p, int cont){
    FILE *fptr=NULL;
    int i=0, z=0;
    if((fptr=fopen("Etapa2.bin", "rb"))==NULL)
    {
        system("cls");
        printf("\n Erro na abertura do arquivo.....\n\n\n");
        system("pause");
    }
    else

```



```

{
    system("cls");
    for(i=0;i<cont;i++)
    {
        fseek(fp_ptr,i*sizeof(livro),0);
        fread(p,sizeof(livro),1,fp_ptr);
        printf("\n\n-----Livro: %i - Titulo: %s - Autor: %s-----", p->reg_livro,
p->titulo, p->autor);
        for(z=0;z<2;z++)
        {
            if((p->status+z)->sigla=='L')
            {
                printf("\n|Status %i |          | %c | RA: ----- | Retirada: --/-- | Devolucao: --/--|\n",
z+1, (p->status+z)->sigla);
            }
            else
            {
                printf("\n|Status %i |          | %c | RA: %s | Retirada: %i/%i | Devolucao: %i/%i|\n",
z+1, (p->status+z)->sigla, (p->status+z)->RA, (p->status+z)->dia_ret, (p->status+z)->mes_ret,
(p->status+z)->dia_dev, (p->status+z)->mes_dev);
            }
        }
    }
    fclose(fp_ptr);
    printf("\n\n\n\n");
    system("pause");
}
}

```

```

void mostra_livro_Parcial_Status(livro *p, int cont){

```

```

    char aux;
    int i=0, z=0, y=0, x=0;
    FILE *fp_ptr=NULL;
    do{
        system("cls");
        printf("\n Status procurado <L/E/R>:");
        scanf("%c", &aux);
        fflush(stdin);
    }while(aux!='L'&&aux!='I'&&aux!='E'&&aux!='e'&&aux!='R'&&aux!='r');
    aux=toupper(aux);
    if((fp_ptr=fopen("Etapa2.bin","rb"))==NULL)

```

```

{
    system("cls");
    printf("\n Erro na abertura do arquivo.....\n\n\n");
    system("pause");
}
else
{
    system("cls");
    for(i=0;i<cont;i++)
    {
        fseek(fp, i*sizeof(livro), 0);
        fread(p, sizeof(livro), 1, fp);
        y=0;
        for(z=0;z<2;z++)
        {
            if((p->status+z)->sigla==aux)
            {
                if(y==0)
                {
                    printf("\n\n-----Livro: %i - Titulo: %s - Autor: %s-----",
p->reg_livro, p->titulo, p->autor);
                }
                y++;
                if(aux=='L')
                {
                    printf("\n|Status %i |          | Sigla: %c | RA: ----- | Retirada: --/-- | Devolucao:
--/--|\n", z+1, (p->status+z)->sigla);
                }
                else
                {
                    printf("\n|Status %i |          | Sigla: %c | RA: %s | Retirada: %i/%i | Devolucao:
%i/%i|\n", z+1, (p->status+z)->sigla, (p->status+z)->RA, (p->status+z)->dia_ret,
(p->status+z)->mes_ret, (p->status+z)->dia_dev, (p->status+z)->mes_dev);
                }
                x=1;
            }
        }
    }
}
if(x==0)
{
    system("cls");
    printf("\n Erro, nao foram encontrados livros com este status .....");
}
}

```

```

        fclose(fptr);
        printf("\n\n\n\n");
        system("pause");
    }
}

```

```

void mostra_livro_Parcial_Titulo(livro *p, int cont){

```

```

    char aux[80], aux2[80];
    int i=0, z=0, x=0;
    FILE *fptr=NULL;

    system("cls");
    printf("\n Titulo do livro procurado: ");
    gets(aux);
    fflush(stdin);
    tratamento(aux, aux);
    if((fptr=fopen("Etapa2.bin","rb"))==NULL)
    {
        system("cls");
        printf("\n Erro na abertura do arquivo.....\n\n\n");
        system("pause");
    }
    else
    {
        for(i=0;i<cont;i++)
        {
            fseek(fptr,i*sizeof(livro),0);
            fread(p,sizeof(livro),1,fptr);
            tratamento(p->titulo, aux2);
            if(strcmp(aux,aux2)==0)
            {
                system("cls");
                printf("\n\n-----Livro: %i - Titulo: %s - Autor: %s-----",
p->reg_livro, p->titulo, p->autor);
                for(z=0;z<2;z++)
                {
                    if((p->status+z)->sigla=='L')
                    {
                        printf("\n|Status %i |          | Sigla: %c | RA: ----- | Retirada: --/-- | Devolucao:
--/--|\n", z+1, (p->status+z)->sigla);
                    }
                    else

```

```

        {
            printf("\n|Status %i |          | Sigla: %c | RA: %s | Retirada: %i/%i | Devolucao:
%i/%i|\n", z+1, (p->status+z)->sigla, (p->status+z)->RA, (p->status+z)->dia_ret,
(p->status+z)->mes_ret, (p->status+z)->dia_dev, (p->status+z)->mes_dev);
        }
    }
    x=1;
}
if(x==0)
{
    system("cls");
    printf("\n Titulo nao encontrado.....\n\n\n");
}
}
fclose(fp);
printf("\n\n\n\n");
system("pause");
}
}

```

```

void tratamento(char *um, char *dois){
    int i=0;
    while(*(um+i)!='\0')
    {
        *(dois+i)= toupper(*(um+i));
        i++;
    }
    *(dois+i)='\0';
}

```