Escreva um programa em C para uma agenda de contados multiusuário. A tela inicial da agenda deve possuir as seguintes opções:

a) Primeiro acesso: esta opção deve ser usada pelos usuários que ainda não estão cadastrados no sistema. Ao selecionar essa opção, o usuário deve ser direcionado ao uma tela onde é possível fazer o seu cadastro. Os dados de um usuário são: `id`, `login` e `nome`. O `id` do usuário é um número sequencial gerado automaticamente pelo sistema e exibido na tela, o usuário deve fornecer apenas seu login e nome. Não permita o cadastro de dois usuários como o mesmo `login`. Se o cadastro for realizado com sucesso, o usuário deve ser direcionado para a tela inicial de sua agenda (descrita posteriormente);

b) Acessar a agenda: para acessar a agenda o usuário deve fornecer o seu `login`, caso o `login` seja válido, deve ser direcionado para a tela inicial da sua agenda (descrita posteriormente);

c) Sair do programa: esta opção encerra o programa.

Na tela inicial da agenda de um usuário deve aparecer um cabeçalho com o texto: “Agenda de <nome-do-usuário>”, onde <nome-do-usuário> é o nome do usuário que acessou a agenda. Abaixo do cabeçalho deve aparecer um menu com as opções:

a) Novo contato: esta opção permite ao usuário cadastrar um novo contato. Os dados de um contato são: `id_usuario`, `id_contato`, `nome`, `telefone` e `email`. O `id_usuario` é o id do usuário ao qual o contato pertence e o `id_contato` é um número sequencial gerado automaticamente pelo sistema;

b) Alterar contato: esta opção permite ao usuário alterar os dados de um contato, para isso o usuário deve informar o id do contato (`id_contato`), caso o id seja válido e seja um contato desse usuário, o sistema mostra os dados atuais do contato e permite a alteração seu nome, telefone e e-mail. Caso o id seja inválido, o sistema exibe uma mensagem de erro;

c) Listar contatos: esta opção mostra no formato de uma tabela todos os contatos do usuário cadastrados. As colunas da tabela devem ser: id do contato (`id_contato`), nome (`nome`), telefone (`telefone`) e e-mail (`email`);

d) Consultar contatos: está opção permite ao usuário consultar seus contatos pelo nome. O usuário deve fornecer parte do nome do contato, e o sistema deve exibir no formato de uma tabela o id, nome, telefone e e-mail de todos os contatos que contenham esse valor em seu nome.

e) Sair da agenda: ao selecionar essa opção o usuário deve voltar para a página inicial da aplicação.

O sistema deverá gravar os dados dos usuários e contatos em arquivos binários. Forneça a implementação dos módulos definidos pelos arquivos *usuario.h* e *agenda.h* mostrados a seguir. A função *main()* da aplicação deve ficar em um arquivo à parte.

```
/* usuario.h */

#ifndef _USUARIO_H
#define _USUARIO_H
#include <stdbool.h>

/* nome do arquivo aonde os usuários são cadastrados */
#define ARQ_USUARIO "usuario.dat"

/* estrutura que representa um usurio */
typedef struct {     int id; // id do
usuário     char login[21]; // login do
usuário     char nome[51]; // nome do
usuário } usuario;

/* função que cadastra um novo usuário no arquivo e direciona-o
* para a tela inicial de sua agenda, caso o cadastro tenha sido
* bem-sucedido */void cadastrar_usuario(void);

/* função que solicita o login do usuário e caso ele seja válido,
* direciona-o para a tela inicial de sua agenda */void
  autenticar_usuario(void);

/* função que devolve verdadeiro caso um usuário com o login
* fornecido como primeiro argumento da função já esteja cadastrado
* e armazena os dados desse usuário na estrutura cujo endereço
* é passado como segundo argumento da função. Se o usuário não *
  estiver cadastrado, a função devolve falso.
* Esta função deve ser usada pelas funções cadastrar_usuario()
* e autenticar_usuario() */
bool usuario_existe(const char *, usuario *);

#endif
```

---

```
/* agenda.h */

#ifndef _AGENDA_H
#define _AGENDA_H

#include <stdbool.h>
#include "usuario.h"

/* nome do arquivo aonde os contados da agenda são cadastrados */
#define ARQ_AGENDA "agenda.dat"

/* estrutura que representa um contato da agenda */
typedef struct {
    int id_usuario; // id do usuário da agenda
```

```
    int id_contato; // id do contato     char
nome[51]; // nome do contato     char
telefone[16]; // telefone do contato     char
email[31]; // e-mail do contato } contato;

/* função que recebe o um usuário como argumento e
* exibe a tela inicial de sua agenda */void
  menu_agenda(usuario);

/* função que cadastra um novo contato para o usuário
* cujo id é fornecido como argumento da função
  */void cadastrar_contato(int);

/* função que recebe como argumento o id de um contato
* e permite alterar os dados desses contato,
  caso seu
* id seja válido */void alterar_contato(int);

/* função que lista todos os contatos de um usuário cujo
* id é fornecido como argumento */void
  listar_contatos(int);

/* função que permite consultar por nome os contatos de
* um usuário cujo id é fornecido como argumento
  */void consultar_contatos(int);

#endif
```