

## TRABALHO 3

### LISTAS ENCADEADAS

Implemente um sistema em C para controle de agendamento de consultas de pacientes em uma clínica médica. O sistema deve atender aos seguintes requisitos:

- Deve ser possível cadastrar **médicos**: incluir médicos e listar todos os médicos.
  - Sobre cada médico, guardar: CRM, nome, área de especialidade e telefone.
- Deve ser possível cadastrar **pacientes**: incluir pacientes e listar todos os pacientes.
  - Sobre cada paciente, guardar: CPF, nome e telefone.
- Deve ser possível **agendar consulta** de um paciente com um médico: agendar consulta, desmarcar consulta e listar todas as consultas.
  - Sobre cada consulta, guardar: paciente, médico, data/horário, convênio a ser utilizado, um flag/status que informa que a consulta está “*agendada*” e uma descrição textual a ser preenchida no momento da consulta (ver restrição 3).
- Deve ser possível **consultar** desde que tenha sido feito um agendamento prévio desta consulta: neste caso, o flag/status que informa que a consulta está agendada deve ser trocado para “*executada*”.

Restrições:

- 1) Uma consulta só pode ser agendada com um determinado médico se houver horário livre na agenda deste médico (considere que as consultas são marcadas a cada 30 minutos) no período das 8h às 12h e das 14h às 18h (o sistema não pode permitir duas consultas com o mesmo médico no mesmo horário);
- 2) Um mesmo paciente não pode ter mais de uma consulta agendada para o mesmo dia e horário, mesmo que com médicos diferentes;
- 3) Cada consulta gera uma descrição textual, onde o médico pode informar quais os exames foram solicitados, quais os medicamentos foram prescritos e outras informações relevantes;
- 4) Para excluir (desmarcar) uma consulta, são fornecidos como entrada o CPF do paciente e o dia/horário da consulta que deseja desmarcar; neste caso, o horário desmarcado fica disponível para um novo agendamento;
- 5) Uma consulta só pode ser agendada por um cliente já cadastrado e com um médico já cadastrado;
- 6) Não pode haver duplicidade de informações sobre médicos e pacientes. Por exemplo, uma consulta não armazena/replica os dados de pacientes e médicos, mas guarda apenas ponteiros que apontam para os respectivos paciente e médico já previamente cadastrados.

Faça um menu de opções que permita ao usuário realizar as operações descritas acima e gerar os seguintes relatórios:

- R1) Listar todas as consultas agendadas para um certo dia (o dia é informado pelo usuário);
- R2) Listar todas as consultas já realizadas por um paciente (o paciente é informado pelo usuário);
- R3) Listar as descrições textuais de uma determinada consulta (o usuário informa o paciente e data/horário da consulta desejada);
- R4) Listar todos nomes de pacientes que consultaram na clínica com os médicos de uma determinada especialidade num determinado mês (especialidade e mês são informados pelo usuário);
- R5) Para cada médico cadastrado, listar o nome de todos os pacientes que já consultaram com ele na clínica.

### Observações para este trabalho:

- **data da entrega: 06 de outubro;**
- os dados devem ser armazenados obrigatoriamente em listas encadeadas;
- podem usar listas simplesmente ou duplamente encadeadas;
- deve ter um menu de opções que permita ao usuário escolher qual ação deseja executar;
- as *structs* devem sempre ser alocadas dinamicamente;
- **o código deve ser organizado em diferentes arquivos e funções (sugestão: *main.c*, *lista.c*, *lista.h*, *menu.c*, *menu.h*);**
- trabalhos atrasados não serão corrigidos - se você não terminou tudo a tempo, entregue o que conseguiu fazer;
- formato da entrega: enviar os arquivos fonte .c e .h (não anexar arquivos executáveis!) para [deise@inf.ufsm.br](mailto:deise@inf.ufsm.br);
- trabalho em grupos de até 2 alunos. Enviar um e-mail por dupla, identificando a disciplina, nome dos alunos e número do trabalho (trabalho 3, por exemplo);
- caso não receba confirmação do recebimento do e-mail em 24hs, contate novamente o professor (reclamações posteriores sobre possíveis problemas no envio do e-mail não serão aceitas);
- cópias da internet e/ou colegas de outras duplas anulam a nota do trabalho de ambos os grupos;
- trabalhos com erros de compilação não serão corrigidos;
- professor usará o Dev-C ou Code Blocks para corrigir os trabalhos. Podem usar qualquer IDE para implementar o trabalho, mas certifiquem-se de que os arquivos estão compilando corretamente em alguma destas IDEs;
- obrigatoriamente a solução de vocês deve fazer uso de funções e passagem de parâmetros (eventuais respostas que pareçam corretas na execução, mas não usem funções e passagem de parâmetros, serão consideradas erradas);
- **não devem ser usadas variáveis globais;**
- a qualquer momento, vocês podem ser chamados para responder questionamentos sobre o trabalho entregue. Quando chamados, os alunos devem apresentar/responder aos questionamentos do professor, os quais farão parte da nota do trabalho;
- juntamente com o trabalho deve ser entregue um arquivo *readme.txt* que descreve sucintamente o que foi e o que não foi feito do enunciado do trabalho;
- peso do trabalho: verificar no site da disciplina em “Avaliação”.