



Universidade Federal de Uberlândia  
Campus Monte Carmelo  
Curso de Bacharelado em Sistemas de Informação

### **Trabalho Prático de Algoritmos e Programação II - 20 pontos**

2º Semestre de 2023    Data de entrega: 13 de abril de 2024    Horário limite: 23:59 horas

#### **• Instruções Gerais para Entrega**

1. Antes de enviar o seu trabalho, considere com atenção e cuidado as instruções aqui contidas. Não seguí-las pode te levar a receber o conceito mínimo (zero).
2. O trabalho deverá ser realizado em dupla ou trio (não serão aceitos trabalhos individuais ou em grupos de mais de três alunos). Não deve haver troca de informações entre grupos distintos. Trabalhos iguais, conceito mínimo para ambos os grupos.
3. A entrega do trabalho será feita via Moodle por meio de link, que será disponibilizado uma semana antes do prazo final.
4. As implementações deverão ser apresentadas em detalhes por todos os integrantes do grupo, nos dias 15 e 17 de abril de 2024. Trabalhos entregues e não apresentados receberão o conceito 0 (zero).
5. Para a apresentação, todos os alunos devem estar preparados para discutir detalhes do código, bem como para justificar as decisões tomadas no decorrer da implementação.
6. O envio deve ser realizado até a data e horário limite estabelecidos para consideração do trabalho. Entregas com atraso de 1 minuto até 24 horas serão considerados com peso de 75%. Não serão aceitas entregas com atraso superior a 24 horas.
7. O trabalho deve ser implementado na linguagem C, e será executado para fins de correção utilizando-se o CodeBlocks.
8. É obrigatória a utilização de indentação no código.
9. Todos os arquivos essenciais do trabalho devem estar contidos em uma pasta com o nome dos autores (sem espaços ou acentos), e esta pasta deve ser compactada e um único arquivo submetido (por exemplo, a entrega pode ser um arquivo nomeado MariaPereira\_JoaoSilva.zip para os alunos Maria Costa Pereira e João Braga Silva, que contém uma pasta nomeada Maria-Pereira-JoaoSilva). Não inclua os arquivos compilados, apenas os códigos fonte e outros arquivos que possam ser necessários para a compilação (e.g., arquivos de cabeçalho).
10. A pasta pode conter um arquivo README.txt com todas as informações que julgar necessárias para a compilação e ou correção do seu trabalho.

#### **• Critérios de Avaliação**

1. Compilação sem erros (utilizarei o CodeBlocks, sistema operacional Windows 10).
2. Execução sem erros (efetue muitos testes para verificar a presença de erros em tempo de execução).
3. Lógica utilizada na implementação.
4. Estruturas implementadas corretamente.
5. Todas as funções devem ser implementadas manualmente.
6. Completude do programa apresentado considerando os requisitos.

## • Trabalho Prático

1. O trabalho consiste na implementação de um programa para gerenciar uma clínica de especialidade escolhida pelo grupo. Devem ser gerenciados os pacientes, os médicos e os atendimentos, e deve ser criada uma estrutura (struct) para cada um deles. As estruturas base são descritas a seguir:
  - Paciente: cpf, nome, telefone, endereço, data de nascimento.
  - Médico: CRM (string), carga horária (número inteiro maior ou igual a 0), salário (número real maior ou igual a 0), especialidade.
  - Atendimento: cpf do paciente, CRM do médico, data do atendimento.
2. O programa deve utilizar vetores de estruturas para cada uma das estruturas base, e estas devem ser salvas em arquivos (separadamente, um arquivo por vetor de estrutura). Os vetores podem ter no máximo 30 registros.
3. Deve haver persistência dos dados: no início do programa, os dados dos pacientes, médicos e atendimentos devem ser carregados para seus vetores a partir dos arquivos para que o programa funcione corretamente com os dados inseridos anteriormente; os dados de pacientes, médicos e atendimentos devem ser salvos nos arquivos para utilização na próxima execução do programa.
4. O programa deverá apresentar um menu que permita que o usuário:
  - Cadastre, altere, consulte ou remova um médico.
  - Cadastre, altere, consulte ou remova um paciente.
  - Cadastre, altere ou remova um atendimento.
  - Liste os dados de todos os médicos cadastrados
  - Liste os dados de todos os pacientes cadastrados
  - Mostre os dados de todos os atendimentos de determinado paciente a partir do seu nome, incluindo o nome do médico, o CRM e a data do atendimento.
  - Gere um relatório de atendimentos no período (criar um arquivo de relatório contendo as informações de todos os atendimentos feitos a partir de uma data definida pelo usuário até o dia atual, incluindo CRM e nome do médico, cpf e nome do paciente, e a data do atendimento).
  - Sair do programa.
5. Todas as alterações e consultas devem ser feitas a partir de uma escolha do usuário para algum campo (exemplos: o usuário digita o nome do paciente que ele quer alterar; o usuário digita o CRM do médico que ele quer buscar etc).
6. As restrições a seguir devem ser atendidas no programa:
  - O programa não pode permitir que um novo cadastro de paciente seja efetuado se o CPF do paciente já estiver cadastrado.
  - Se o CRM do médico já estiver cadastrado, o novo cadastro não deve ser efetuado.
  - Não pode haver mais de um atendimento entre um médico e um paciente no mesmo dia. Em dias diferentes não há problema.
  - Só pode haver atendimento de pacientes e médicos já cadastrados.
  - Um paciente ou médico não pode ser removido se fizer parte de algum atendimento.
7. As restrições mencionadas também não podem ser violadas em caso de alterações.
8. Lembre-se de controlar o número de registros já feitos em cada estrutura, e que esse número não pode ultrapassar o tamanho do vetor.
9. O programa principal deve conter apenas o suficiente para o funcionamento do menu e das estruturas. Todas as outras funcionalidades devem ser implementadas por meio de funções.
10. **DICA:** crie uma função de busca para cada estrutura que retorna o índice do elemento encontrado, e chame esta busca sempre que necessário nas outras funções.