Centro Federal de Educação Tecnológica de Minas Gerais - Departamento de Computação

Laboratório de Programação de Computadores I (2ECOM.002)

Trabalho prático 2

Professor: Amadeu Almeida **Data de entrega:** 6 de Dezembro de 2019

Instruções: leia com atenção **todas** as orientações abaixo antes de começar o trabalho.

- Esta atividade deve ser feita em grupos de 1 à 3 pessoas e vale 15 pontos na disciplina teórica e 20 na prática.
- Faça esta tarefa no **CodeBlocks** ou em algum software similar e submeta no SIGAA o código fonte em **C** com as soluções do trabalho.
- Caso a tarefa seja feita em dupla ou em trio, apenas um aluno precisa enviá-la. Porém, os nomes de **todos** os integrantes devem estar comentados na primeira linha do programa.
- O prazo de entrega é até às 23:59 do dia 6 de Dezembro.
- Códigos fonte em PDF e submissões em atraso serão desconsiderados.
- Qualquer indício de cópia resultará no **anulação** de todo o trabalho. É importante ressaltar que todos os códigos serão inseridos em um detector de plágio que identifica os arquivos duplicados.
- O enunciado deste trabalho possui 3 páginas.

Durante o ano de 2019, o sistema de cadastro dos alunos de uma escola apresentou inúmeros problemas técnicos e de estabilidade. Preocupados com a maneira que tais inconvenientes têm afetado a rotina estudantil e acadêmica, a diretoria desta instituição decidiu que este programa será substituído em um futuro próximo e contratou vocês para criarem um protótipo de um novo software composto por:

- Uma struct chamada **Aluno**
- Cinco funções cujos nomes são:
  - main
  - adicionarAluno
  - editarAluno
  - removerAluno
  - imprimirAlunos

struct Aluno

A **struct Aluno** é um registro que armazena as informações de um discente da escola. Ela possui os seguintes campos:

- int numeroMatricula - é um número que contém 6 algarismos (entre 100000 e 999999).
- char nome[26]
- char sobrenome[26]
- int anoNascimento - deve ser um número entre 1900 e 2019.
- char curso[26]

  Observação: o nome, o sobrenome e o curso de um aluno possuem no máximo 25 caracteres.

Função main

A função **main** possui dois objetivos:

1. Criar as variáveis que são utilizadas nas outras funções do programa.
2. Direcionar o fluxo do sistema para os outros métodos conforme a opção selecionada pelo usuário.

As seguintes variáveis devem ser declaradas neste método:

1. **struct aluno listaAlunos[100]** - vetor de registros que armazena as características de todos os alunos listados no sistema.

Observação: cada posição do vetor contém os dados de um aluno distinto.

2. **int contadorAlunos = 0;** - variável que contabiliza a quantidade de discentes cadastrados no vetor **listaAlunos**.

Observação: o software comporta a inclusão de no máximo 100 alunos.

O corpo da função **main** contém instruções que:

1. Chamam os outros métodos do programa de acordo com a opção selecionada pelo usuário.
   (a) Por exemplo, se o utilizador escrever 1, o sistema chama a função **adicionarAluno** e inicia o processo de inclusão dos dados de um discente.
2. Atualizam o contador de alunos sempre que um cadastro ou uma remoção ocorrer com sucesso. Nestes casos, as funções **adicionarAluno** e **removerAluno** retornam o valor 1 para a **main**.
3. Terminam a execução do programa, caso uma opção inválida seja digitada.

Função adicionarAluno

O objetivo da função **adicionarAluno** é cadastrar um novo aluno no sistema. O cabeçalho deste método é **int adicionarAluno(struct aluno listaAlunos[], int quantidadeAlunosNoRegistro)** e o corpo dele possui comandos que:

1. Permitam ao usuário digitar as características de um aluno, introduzindo-as nos campos de uma das posições do vetor de registros.
2. Imprimem mensagens explicando ao utilizador qual é a informação que ele deve inserir no programa.
   (a) **Exemplo:** printf("Digite um numero de matricula entre 100000 e 999999: \n");
3. Impossibilitem a inclusão de novos dados caso o vetor de registros contenha 100 alunos.
4. Garantem as restrições de:
   (a) Número de matricula (6 algarismos).
   (b) Ano de nascimento (entre 1900 e 2019).

   Observação: caso o usuário digite um número de matricula ou um ano de nascimento inválido, o sistema deve solicitar que ele reescreva este valor.
5. Retornam 1 caso as informações de um aluno sejam adicionadas com sucesso ou 0, caso contrário.

Função editarAluno

O método **editarAluno** ambiciona alterar uma ou mais características de um aluno cadastrado no sistema. O cabeçalho desta função é **void editarAluno(struct aluno listaAlunos[], int contadorAlunos)** e o seu corpo tem instruções que:

1. Comparam o número de matrícula informado pelo usuário com aqueles que estão armazenados no vetor contendo os dados dos alunos. Posteriormente é impresso:

   (a) Um menu perguntando qual é o campo que será alterado, caso a matrícula pertença a algum aluno.

   (b) Uma mensagem comunicando que o número inserido não pertence à nenhum discente.

   Observação: caso a matricula inserida seja inválida, o programa deve voltar para a função **main**.

2. Possibilitem ao utilizador editar uma ou mais informações de um aluno, alterando-as nos campos de seu registro dentro do vetor.

   Observação: apenas uma característica de um aluno pode ser alterada por vez. Neste caso, o fluxo do programa deve voltar ao menu desta função após a modificação do campo.

3. Não violem as restrições de:

   (a) Número de matricula (6 algarismos).

   (b) Ano de nascimento (entre 1900 e 2019).

   Observação: se o usuário informar um número de matricula ou um ano de nascimento inválido, o programa tem que pedir para ele reescrever este valor.

4. Termine a execução desta função quando o usuário digitar uma opção inválida.

   Observação: este método não devolve nenhum valor para a **main**.

Função removerAluno

A função **removerAluno** objetiva eliminar todas as informações referentes a um aluno a partir da digitação de seu número de matrícula. O cabeçalho deste método é **int removerAluno(struct aluno listaAlunos[], int contadorAlunos)** e o corpo dele deve conter comandos que:

1. Comparam o número de matricula digitado pelo utilizador com aqueles que se encontram no vetor que armazena os registros dos alunos e:

   (a) Removem os dados de um aluno, caso sua matrícula seja igual à informada.

   (b) Imprimem uma mensagem avisando que o número inserido não está no vetor.

   Observação: em ambos os casos, a função deve terminar sua execução e voltar para o método **main**.

2. Devolvem 1 se as informações de um aluno forem deletadas com sucesso ou 0 caso o número de matricula informado seja inválido.

Função imprimirListaAlunos

O proposito do método **imprimirListaAlunos** é imprimir os dados de todos os alunos que estão cadastrados no programa. O cabeçalho desta função é **void imprimirListaAlunos(struct aluno listaAlunos[], int contadorAlunos)** e o seu corpo possui instruções que:

1. Imprimem todas as informações dos alunos que se encontram no vetor de registros.

Observação: este método não retorna nenhum valor para a **main**.