

**ORGANIZAÇÃO DE ESTRUTURAS DE ARQUIVOS**  
**PROVA 1 - 2019.1**  
**PROF. RENATO CAMPOS MAURO**

Nome: \_\_\_\_\_

Muitos aplicativos geram ou lêem arquivos no formato CSV. CSV é sigla para “Comma Separated Values” (valores separados por vírgula). Trata-se de um arquivo texto, onde cada linha do arquivo representa um registro. O trecho abaixo é um exemplo de um arquivo no formato CSV. Observe que a primeira linha contém o nome das colunas.

```
id_inscricao;curso;cpf;dataNascimento;sexo;email;opcaoQuadro
1;ADMINISTRAÇÃO;093.157.067-06;1981-01-26;M;eanes25@gmail.com;A
2;ADMINISTRAÇÃO;118.678.527-64;1986-06-11;M;tbn2007.tb@gmail.com;B
3;ELETRÔNICA;148.197.787-30;1993-02-01;M;lucasmoreira@poli.ufrj.br;A
4;ELETROTÉCNICA;127.297.057-40;1991-06-06;M;wpaulo28@yahoo.com.br;A
...
18;EDIFICAÇÕES;430.794.317-15;1957-01-29;M;toninhochavessouza@yahoo.com.br;C
19;EDIFICAÇÕES;040.105.135-86;1997-01-12;M;neto13fjv@gmail.com;A
20;MECÂNICA;166.770.527-02;1999-05-05;M;rochafabricio38@gmail.com;A
```

Este arquivo é uma amostra dos inscritos para um concurso do CEFET que acontece anualmente. O Arquivo chama-se candidatos.csv.

**Questão 1** O arquivo do exemplo possui registros de tamanho fixo ou variável? Quais são as vantagens e desvantagens desta representação? (0,5 ponto)

**Questão 2** . Escreva um programa que receba como argumento do comando de linha um CPF e imprima na tela o nome do curso para o qual o candidato se inscreveu. Faça na linguagem de programação de sua preferência. Se for em Java, suponha que já exista um classe Registro, com gets e sets para cada atributo e a função leRegistro. Se for em C, considere que já exista a função leRegistro, que lê uma linha do arquivo e separa os campos em cada um dos campos da estrutura. (1,5 pontos)

```
struct Registro {
    int id_inscricao;
    char curso[20];
    char cpf[15];
    char dataNascimento[11];
    char sexo;
    char email[40];
    char opcaoQuadro;
};

/*****
Le e decodifica a linha do arquivo
*f e armazena no registro *r.
retorna 1 = OK ou 0 = EOF
*****/
int leRegistro(Registro *r, FILE *f);
```

**Questão 3** Considere agora dois arquivos no mesmo formato: candidatosA.csv e candidatosB.csv, com candidatosA.csv ordenado por CPF e candidatosB.csv não ordenado. Faça um programa que imprima na tela o email dos candidatos que se candidataram para as duas edições do concurso, ou seja, aqueles que aparecem nos dois arquivos simultaneamente. Escreva de forma eficiente! (1,5 pontos)

**Questão 4** Criação de um índice para pesquisa no arquivo dos candidatos: Suponha que o arquivo candidatoA tenha exatamente 10000 registros. Escreva um programa que leia cada uma das linhas do arquivo, armazene cada registro em uma estrutura de dados em memória (um array em C ou um ArrayList em java), ordene e salve os dados desse arquivo em um novo arquivo chamado indiceA.dat. Suponha que as funções para leitura e escrita dos registros do índice já existam, nos mesmos moldes das questões anteriores, portanto não precisam ser implementadas. Use a função de ordenação nativa da linguagem de programação de sua preferência. Explique como este arquivo pode ser utilizado para fazer pesquisas através do CPF de forma eficiente. (1,5 pontos)

```
struct Indice {  
    char cpf[15];  
    long posicaoNoCSV;  
};
```

**Questão 5** Em um MAXHEAP inicialmente vazio, inclua as chaves 82, 31, 28, 4, 45, 27, 59, 79, 35. Depois remova cada uma delas. Desenhe passo a passo a estrutura de dados. (1,0 ponto)

BOA PROVA!!