



## Lista de exercícios 6 – Arquivos – **Prazo: combinado em aula**

### INSTRUÇÕES:

- A – Todos os arquivos compactados em formato ZIP (.zip). Inclua apenas os códigos-fonte (.c e .h), ou seja, não me envie os executáveis (.exe). O zip deve seguir o padrão: *[SEU\_NOME].zip*.
- B – Todas as funções solicitadas na lista de exercícios devem estar dentro de sua biblioteca. A biblioteca deve seguir o padrão: *[SEU\_NOME].h* para os protótipos e *[SEU\_NOME].c* para a descrição das funções.
- C – Os códigos-fonte dos exercícios devem conter APENAS a função `main()` e devem seguir o padrão: *ex[NUMERO].c*.
- D – A lista de exercícios é entregue pelo Google Classroom.

### PARTE 1 – PUTC e GETC

Considere que existe um arquivo chamado `exemplo.txt` com o seguinte conteúdo:

```
Hello  
World!
```

- 1 – Escreva a função `int nLinhas(char * nomeArq)` para retornar quantas linhas o arquivo `nomeArq` possui. Exemplo:
- ```
nLinhas("exemplo.txt") = 2
```
- 2 – Escreva a função `int contaVogais(char * nomeArq)` para retornar quantas vogais o arquivo `nomeArq` possui. Exemplo:
- ```
contaVogais("exemplo.txt") = 3
```
- 3 – Escreva a função `int contaVogaisOuConsoantes(char * nomeArq, int opcao)` para retornar quantas vogais (`opcao=0`) ou consoantes (`opcao=1`) o arquivo `nomeArq` possui. Considere utilizar a função `int contaVogais(char * nomeArq)` desenvolvida anteriormente. Exemplo:
- ```
contaVogaisOuConsoantes("exemplo.txt", 1) = 7 //consoantes  
contaVogaisOuConsoantes("exemplo.txt", 0) = 3 //vogais
```
- 4 – Escreva a função `int contaCaracter(char * nomeArq, char ch)` para retornar quantas ocorrências do caracter `ch` no arquivo `nomeArq`. Exemplo:
- ```
contaCaracter("exemplo.txt", 'l') = 3  
contaCaracter("exemplo.txt", '!') = 1  
contaCaracter("exemplo.txt", '\n') = 1
```

### PARTE 2 – FPUTS e FGETS

- 5 – Escreva a função `void relatorioChar(char * arqFonte, char * arqDestino)` que escreve em `arqDestino` quantas vezes cada letra do alfabeto apareceu dentro do `arqFonte` possui. Considere utilizar a função `int contaCaracter(char * nomeArq, char ch)` desenvolvida anteriormente. Exemplo:
- ```
relatorioChar("exemplo.txt", "relatorio.txt")
```

Arquivo `relatório.txt`:

```
a=0; b=0; c=0; d=0; e=1; ... w=1; x=0; y=0; z=0;
```

6 – Escreva a função `void catArqs(char * arq1, char * arq2, char * arqResult)` que concatena em um único arquivo `arqResult` o conteúdo do `arq1` e `arq2`. Considere que o nome dos arquivos podem ser iguais. Exemplo:

```
catArqs("exemplo.txt", "relatorio.txt", "exemplo.txt")
```

Arquivo exemplo.txt:

Hello

World!

```
a=0; b=0; c=0; d=0; e=1; ... w=1; x=0; y=0; z=0;
```

7 – Escreva a função `int findWord(char * arquivo, char * palavra)` para encontrar a primeira ocorrência de uma palavra em um arquivo. A função retorna a linha do arquivo em que a palavra foi encontrada. Caso palavra não seja encontrada, retorna -1. Saída esperada:

```
findWord("exemplo.txt", "World")=2
```

```
findWord("exemplo.txt", "Programacao")=-1
```

8 – Escreva a função `int countWord(char*, char*)` para contar o número de ocorrências de palavra em um arquivo. Saída esperada:

```
countWord("exemplo.txt", "World")=1
```

```
countWord("exemplo.txt", "; ") =26
```

```
countWord("exemplo.txt", "Programacao")=0
```

9 – Escreva a função `int removeWord(char*, char*)` para remover todas as ocorrências de palavra em um arquivo. Saída esperada:

```
removeWord("exemplo.txt", "; ") =1
```

```
removeWord("exemplo.txt", "Programacao")=0
```

Arquivo exemplo.txt após as duas chamadas de `removeWord()`:

Hello

World!

```
a=0b=0c=0d=0e=1 ... w=1x=0y=0z=0
```

### **PARTE 3 – FPRINTF e FSCANF**

Desenvolva um programa para gerenciar as notas dos alunos de uma turma salva em um arquivo. O programa deve ter um menu contendo as seguintes opções:

(a) Definir informações da turma;

(b) Inserir aluno e notas;

(c) Remover aluno;

(d) Editar nota aluno;

(e) Exibir alunos aprovados;

(f) Exibir alunos reprovados;

(g) Exibir estatísticas (média geral, % reprovados, % aprovados s/ recuperação, % aprovados c/ rec.);

(h) Sair do programa (fim).

Faça a rotina que gerencia o menu dentro do main, e para cada uma das opções deste menu, você vai criar uma função específica.

Considere que o nome do arquivo é determinado pelo o que usuário digitou na opção (a). Assumindo que o usuário gerou TSI\_2AN, o arquivo TSI\_2AN.txt segue o padrão abaixo:

| # | matricula      | nome | nota1 | nota2 | recuperacao |
|---|----------------|------|-------|-------|-------------|
| 1 | Joao da Silva  |      | 7.5   | 3.5   | 9.0         |
| 2 | Maria de Souza |      | 10.0  | 8.0   | 0.0         |

...

10 – *Opção (a)* Escreva a função `char * defineTurma(char * curso, int sem_ano, char turno)` que cria um arquivo com o seguinte padrão: `[CURSO]_[SEMESTRE/ANO]A[TURNO]`. E retorna uma string com o mesmo padrão. Exemplo:

```
defineTurma("TSI", 2, 'N') = TSI_2AN //tambem criou TSI_2AN.txt
defineTurma("INF", 4, 'N') = INF_4AT //tambem criou INF_4AT.txt
```

11 – *Opção (b)* Escreva a função `int inserirAluno(char * turma, char * aluno, float n1, float n2, float rec)` para inserir um aluno no arquivo turma. A função retorna 1 em caso de sucesso e 0/NULL em caso de erro. Sugestão: para gerar a matrícula utilize a função `int nLinhas(char * nomeArq)`. Exemplo:

```
inserirAluno("TSI_2AN", "Ciclano da Silva", 2.0, 4.0, 0.0) = 1
inserirAluno("Turma_aleatoria", "Hacker", 10.0, 10.0, 0.0) = 0
```

12 – *Opção (c)* Escreva a função `int removerAluno(char * turma, int matricula)` para remover um aluno do arquivo turma. A função retorna 1 em caso de sucesso e 0/NULL em caso de erro. Exemplo:

```
removerAluno("TSI_2AN", 2) = 1 //Aluna Maria
removerAluno("TSI_2AN", 50) = 0 //Aluna não existe
removerAluno("Turma_aleatoria", 1) = 0 //Turma não existe
```

13 – *Opção (d)* Escreva a função `int editarAluno(char * turma, int matricula, float n1, float n2, float rec)` para editar as notas de um aluno do arquivo turma. A função retorna 1 em caso de sucesso e 0/NULL em caso de erro. Exemplo:

```
editarAluno("TSI_2AN", 2, 10.0, 10.0, 0.0) = 1 //Aluna Maria
editarAluno("TSI_2AN", 50, 10.0, 10.0, 0.0) = 0 //Aluno não existe
editarAluno("Turma_aleatoria", 1, 0.0, 0.0, 0.0) = 0 //Turma não existe
```

14 – *Opções (e) e (f)* Escreva a função `void exibeAlunos(char * turma, char status)` para imprimir a lista de alunos da turma de acordo com o status. A variável pode assumir os valores 'a', 'b' e 'r' para alunos aprovados sem recuperação, aprovados com recuperação e reprovados, respectivamente. Exemplos:

```
exibeAlunos("TSI_2AN", 'a') //aprovados sem recuperacao
Printando alunos aprovados...
2 Maria de Souza 10.0 8.0 0.0
4 Beltrano Souza 8.0 8.0 0.0
```

```
exibeAlunos("TSI_2AN", 'b') //aprovados com recuperacao
Printando alunos aprovados com recuperacao...
1 Joao da Silva 7.5 3.5 9.0
```

```
exibeAlunos("TSI_2AN", 'r') //reprovados
Printando alunos reprovados...
3 Ciclano da Silva 2.0 4.0 0.0
```

15 – *Opção (g)* Escreva a função `void exibeEstatisticas(char * turma)` para imprimir algumas estatísticas da turma. Exemplo:

```
exibeEstatisticas("Turma_aleatoria")
Turma não existe
```

```
exibeEstatisticas("TSI_2AN")
Media = 7.25 % aprovados sem rec = 50.0
% aprovados com rec = 25.0 % reprovados = 25.0
```