

# Prática 11

Além deste documento, o zip contém o programa pratica11.c e os arquivos CALC.txt, AEDS1.txt e GAAL.txt

# Problemas

Considere que o professor armazena as notas de cada disciplina que leciona em um arquivo diferente (GAAL.txt, AEDS1.txt etc).

Formato dos arquivos: matrícula, nome, nota1, nota2

Considere ainda que o professor pode escolher se deseja ou não que a média das notas seja exibida junto com a listagem do conteúdo do arquivo.

Modifique o programa `pratica11.c` de modo que o nome do arquivo e seu modo de exibição sejam opções definidas como parâmetros de execução do programa.

Modifique o programa `pratica11.c` de modo a permitir que as notas dos alunos possam ser modificadas.

# Modificação de conteúdo de arquivos

- O programa `pratica11.c` inclui uma nova função: `modificarNotas`.
- Modificar as informações contidas em um `arquivo textual` (informações codificadas como caracteres ASCII), `não é uma tarefa simples`.
- Isto porque os arquivos textuais podem ser abertos apenas nos modos `"r"`, `"w"` e `"a"`.
- Se um arquivo textual existente for aberto no modo `"w"`, o conteúdo atual deste arquivo será destruído.

# Modificação de conteúdo de arquivos

- Dessa forma, para modificar informações em um arquivo textual, é preciso usar um **arquivo temporário**.
- A função **modificarNotas** deve utilizar, além do arquivo de nome **nome\_arquivo** (ex: “AEDS1.txt”), um arquivo temporário **temp.txt**.
- O arquivo de nome **nome\_arquivo** é aberto para leitura (“r”) e o arquivo **temp.txt** é aberto para gravação (“w”).
- As informações lidas no arquivo de nome **nome\_arquivo** poderão ser gravadas com ou sem alteração no arquivo **temp.txt**.

# Modificação de conteúdo de arquivos

- Ao final do processamento, o conteúdo do arquivo de nome `nome_arquivo` irá se manter inalterado, mas o arquivo `temp.txt` irá conter as informações atualizadas.
- Como as informações atualizadas devem permanecer no arquivo de nome `nome_arquivo`, é preciso destruir o arquivo original e renomear o arquivo temporário:

```
remove(nome_arquivo);  
rename("temp.txt", nome_arquivo);
```

- As operações realizadas pelas funções `remove` e `rename` só serão realizadas se os arquivos em questão estiverem fechados.

# Parâmetros de um programa

- Quando um programa em C é chamado em uma linha de comando, o sistema operacional passa os argumentos da linha como parâmetros de main.
- Isto pode ser utilizado para estabelecer opções de execução do programa.
- Considere o cabeçalho da função main a seguir:

```
int main(int argc, char *argv[])
```

# Parâmetros de um programa

- Quais são os significados dos argumentos de `main`?

- `argc`: número de argumentos na linha de comando. O primeiro argumento é sempre o nome do programa. Assim, `argc` é um inteiro maior ou igual a 1.
- `argv[ ]`: um vetor de strings contendo cada um dos parâmetros da linha de comando.

- O trecho abaixo imprime o “help” do programa caso não tenham sido passados argumentos

```
if(argc == 1) { //se chamei o a.exe sem parametros:
    printf("\nParametros do programa:");
    printf("\n-d [XXX] ..... Sigla da disciplina. Ex: -d AEDS1");
    printf("\n-m [S ou N] ..... Exibir media. Ex: -m S");
    printf("\n-l [#] ..... Le informacao sobre # alunos. Ex: -l 4");
    printf("\n-e ..... Modifica nota de aluno.");
    return 1;
}
```

# Parâmetros de um programa

- O programa pode ser chamado com as opções `-d`, `-m`, `-l` e `-e`.
- “-d” para indicar a disciplina
- “-m” para mostrar ou não a média das notas
- “-l” para ler a informação de n alunos
- “-e” para editar a nota de um aluno



# Parâmetros de um programa

- O programa pode ser chamado com as opções `-d`, `-m`, `-l` e `-e`.
- “-d” para indicar a disciplina
- “-m” para mostrar ou não a média das notas
- “-l” para ler a informação de n alunos
- “-e” para editar a nota de um aluno

# Parâmetros de um programa

- O programa pode ser chamado com as opções `-d`, `-m`, `-l` e `-e`.
- Estas opções podem ser seguidas por parâmetros
  - `-d SIGLA_DA_DISCIPLINA` (ex: `-d AEDS1`)
  - `-m [S ou N]` (ex: `-m S`)
  - `-l NUM_ALUNOS` (ex: `-l 5`)
  - `-e` (sem parâmetro)
- Assim, uma chamada possível ao programa seria:

```
a.exe -d GAAL -m N -e
```

- Neste caso, `argc = 6`.

# Parâmetros de um programa

- Além disso, o vetor de strings `argv` conterá:

- `argv[0]` = “a.exe”
- `argv[1]` = “-d”
- `argv[2]` = “GAAL”
- `argv[3]` = “-m”
- `argv[4]` = “N”
- `argv[5]` = “-e”

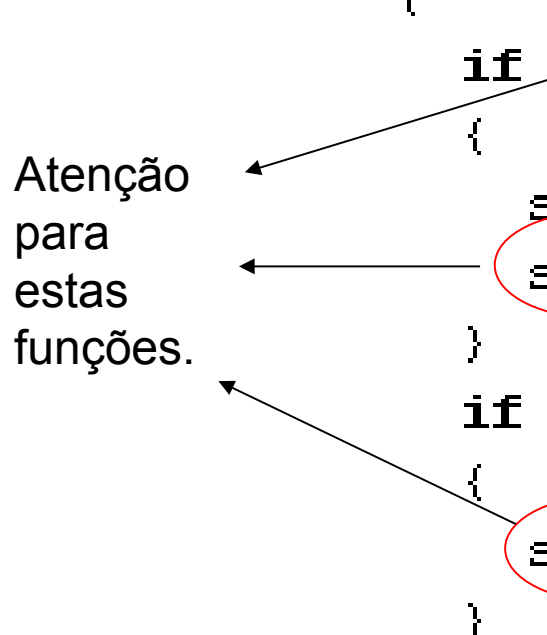
- Portanto, o programa deverá considerar o arquivo de dados correspondente à disciplina `GAAL` (o “`.txt`”) e não deverá exibir a média das notas.
- **Atenção!** A forma de passar parâmetros ao programa poderia ser diferente.

# Parâmetros de um programa

- No programa `pratica11.c`, o tratamento dos parâmetros passados para a função `main` é realizado pelas seguintes instruções:

```
for (i = 1; i < args; i++)
{
    if (strcmp(arg[i], "-d") == 0)
    {
        strcpy(nome_arq, arg[i+1]);
        strcat(nome_arq, ".txt");
    }
    if (strcmp(arg[i], "-m") == 0)
    {
        strcpy(exibir_media, arg[i+1]);
    }
}
```

Atenção  
para  
estas  
funções.



# Parâmetros de um programa

- O significado destas funções é apresentado a seguir:

Função	Significado
strcmp(a,b)	Compara strings <b>a</b> e <b>b</b> , retornando: <b>0</b> , se <b>a</b> for igual a <b>b</b> ; um valor menor do que <b>0</b> , se <b>a</b> for menor do que <b>b</b> (na ordem lexicográfica); um valor maior do que <b>0</b> , se <b>a</b> for maior do que <b>b</b> .
strcpy(a,b)	Atribui o string <b>b</b> ao string <b>a</b> .
strcat(a,b)	Concatena o string <b>b</b> ao string <b>a</b> .

# Desafio para os (muito) fortes!

- Implementar um procedimento que ordena um arquivo pelo número de matrícula.
- O arquivo CALC.txt deve, por exemplo, ficar assim:

```
4,Stannis Baratheon, 8.5, 4.8  
17,Maria Capitolina, 5.0, 4.4  
23,Michael Jordan, 9.0, 9.0  
55,Luke Skywalker, 7.5, 6.1
```