

```

// 202/06/20
// MAC2166 - 2018 - Turmas 13 e 14
// Prof. Leo^nidas de Oliveira Branda~o

// Exemplos para leitura/gravacao de arquivos.
// Este codigo, como esta', supoe que sera' rodado via linha de comando
(abrir terminal
// e digitar algo como 'python exemplo_arquivos.py entrada.txt saida.txt'.
// Se preferir, troque as linhas na "main" com (C) os ... argv

// Exemplo de arquivo para teste: gerar um arquivo texto com as 3 linhas
seguintes
// 2 4
// 1 2 3 4
// 5 6 7 8

// Para compilar com linha de comando gerando codigo para testar alguns erros
em execucao:
// gcc -fsanitize=address -g -o exemplo_arquivos exemplo_arquivos.c

#include <stdio.h>

#include <string.h> // para a funcao 'strcpy(..)'

#define MAXCOMPLINHA 512 // comprimento maximo de cada linha do arquivo
(experimente trocar para 10 entao veja (A) sobre funcao 'fgets')
#define MINP 3 // numero minimo de parametro (nome desse arquivo
e o nome de 2 arquivos de entrada)

#define MAXL 10 // para dimensoes de matriz
#define MAXC 10

// Impressao da matriz
void imprima_matriz_ao_uniforme (int mat[][MAXC], int nl, int nc) {
    int i = 0, j;
    for (i=0; i<nl; i++) {
        // nc = numc[i]; // costum 2 operacoes pegar o valor em 'numc[i]' como
isso e' feito muitas vezes no "j<..." e' mais eficiente variavel auxiliar
'nc'
        for (j=0; j<nc; j++)
            printf("%2d ", mat[i][j]);
        printf("\n");
    }
}

// Leitura de arquivo com uma matriz, tendo na primeir linha o numero de
linhas e de colunas
int leitura_de_numeros (char *arquivol, int mat[][MAXC], int *nl, int *nc) {
    FILE *apont_arq = fopen(arquivol, "r"); // read only
    char linha1[MAXCOMPLINHA]; //ATENCAO: com essa linha, o ultimo lote usado
fica em posicao de memoria alocada provisoriamente aqui!
    int conta = 0, result, aux, i, j;

```

```

if (apont_arq==NULL) {
    printf("Arquivo vazio: %s\n", arquivo1);
    return -1;
}
fscanf(apont_arq, "%d %d", nl, nc); // primeira linha deve ter 2 inteiros
for (i=0; i<*nl; i++)
    for (j=0; j<*nc; j++) {
        result = fscanf(apont_arq, "%d", &mat[i][j]);
        conta++;
    }
return conta;
}

// Gravacao de arquivo com uma matriz, tendo na primeir linha o numero de
linhas e de colunas
int gravacao_de_numeros (char *arquivo1, int mat[][MAXC], int nl, int nc) {
    FILE *apont_arq = fopen(arquivo1, "w"); // "w" = "write"
    char linha1[MAXCOMPLINHA]; //ATENCAO: com essa linha, o ultimo lote usado
fica em posicao de memoria alocada provisoriamente aqui!
    int i, j;
    if (apont_arq==NULL) {
        printf("Sem permissao de gravacao? %s\n", arquivo1);
        return -1;
    }
    fprintf(apont_arq, "Linhas x colunas = %d %d\n", nl, nc); // primeira linha
deve ter 2 inteiros
    for (i=0; i<nl; i++) {
        fprintf(apont_arq, "%2d : ");
        for (j=0; j<nc; j++)
            fprintf(apont_arq, "%d ", mat[i][j]);
        fprintf(apont_arq, "\n");
    }
    return 0;
}

// Leitura de arquivo como linhas ("strings")
// Devolve: -1 => arquivo vazio ou inexistente; -2 => arquivo maior que
capacidade da "string"; -3 => MAXCOMPLINHA insuficiente
int leitura_por_linhas (char *arquivo1, char vet_linhas[][MAXCOMPLINHA]) {
    FILE *apont_arq = fopen(arquivo1, "r"); // read only
    char fimArq1 = 0;
    char linha1[MAXCOMPLINHA];
    int i = 0;

    if (apont_arq==NULL) {
        printf("Arquivo vazio: %s\n", arquivo1);
        return -1;
    }

    while (1) { // elimina linhas iniciais
        if (fgets(linha1, MAXCOMPLINHA, apont_arq)==NULL || ferror(apont_arq) ||
feof(apont_arq) ) { // (A)
            break;
        }
    }
}

```

```

    printf("linha %2d: %s", i, linha1); // em geral a linha vem com '\n',
    entao nao precisa de quebra ao final
    // printf("linha %2d: %s\n", i, linha1); // (A) se MAXCOMPLINHA nao
    comportar a linha, precisa de quebra
    if (i >= MAXL) {
        printf("Erro: o arquivo de entrada tem mais que %d linhas!\n", MAXL);
        return -3;
    }
    strcpy(vet_linhas[i], linha1);
    i++;
}
fclose(apont_arq);
return i;
} // int leitura_por_linhas(char *arquivo1, char
vet_linhas[][MAXCOMPLINHA])

```

```

// Para testar sozinho esta base eliminar o "" do "main" abaixo
int main (int argc, char *argv[]) {
    char dados[MAXL][MAXCOMPLINHA];
    int dados_int[MAXL][MAXC];
    char *nome_arq_entrada, *nome_arq_saida; //L para usar com leitura
    'lerFrase(...) '
    int resp, nl, nc, total, i; //[MAXL]

    if (argc<MINP) { // (C)
        printf("Esperado ao menos %d parametros. ", MINP);
        printf("Encontrado: %d parametro(s)\nLinha de comando: ", argc);
        for (i=0; i<argc; i++) printf("%s ", argv[i]);
        printf("\n");
        return 1;
    }

    nome_arq_entrada = argv[1];
    printf("Nome do arquivo de entrada: %s\n", nome_arq_entrada);
    // resp = leitura_por_linhas(nome_arq_entrada, dados);
    total = leitura_de_numeros(nome_arq_entrada, dados_int, &nl, &nc);
    printf("Total de linhas: %d\n", nl);
    printf("Total de elementos: %d\n", total);
    imprima_matriz_nao_uniforme(dados_int, nl, nc);

    nome_arq_saida = argv[2];
    printf("Nome do arquivo de saida: %s\n", nome_arq_saida);

    resp = gravacao_de_numeros(nome_arq_saida, dados_int, nl, nc);
    if (resp == -1) {
        printf("Erro ao tentar ler o arquivo: %s\n", nome_arq_entrada);
        return 1; // erro!
    }

    return 0;
}

```