

S T Q Q S S D

Ana Carolina da Silva Araujo

Inf-IA

Lista 12

1) #include <stdio.h>

#include <stdlib.h>

```
int main (void) {
```

```
FILE *fp = fopen ("dados.txt", "w");
```

```
if (!fp) {
```

```
perror ("Erro ao criar dados.txt");
```

```
return 1;
```

```
}
```

```
fprintf (fp, "Olá, mundo! Este é meu primeiro  
arquivo em C.\n");
```

```
fclose (fp);
```

```
printf ("Arquivo 'dados.txt' criado com sucesso.\n");
```

```
return 0;
```

```
}
```

2) #include <stdio.h>

#include <stdlib.h>

```
int main (void) {
```

```
FILE *fp = fopen ("dados.txt", "r");
```

```
if (!fp) {
```

```
perror ("Erro ao abrir dados.txt");
```

```
return 1;
```

```
}
```

```
char buf [256];
```

```
while (fgets (buf, sizeof (buf), fp) != NULL) {
```

```
fflush (buf, stdout);
```

```
}
```

S T Q Q S S D

```
fclose(fp);
return 0;
}
```

3) #include <stdio.h>
 #include <stdlib.h>

```
int main (void) {
    FILE *fp = fopen("números.txt", "w");
    if (!fp) { perror("Erro ao criar números.txt");
    return 1;
}
```

```
int n;
printf ("Digite 10 números inteiros:\n");
for (int i=0; i<10; ++i) {
    if (scanf ("%d", &n) != 1) {
        fprintf (stderr, "Entrada invalida\n");
        fclose (fp);
        return 1;
    }
    fprintf (fp, "%d\n", n);
}
fclose (fp);
```

```
fp = fopen ("números.txt", "r");
if (!fp) {
    perror ("Erro ao abrir números.txt");
    return 1;
}
printf ("\nConteúdo de números.txt:\n");
```

mirat

S T Q Q S S D

—V—V—

```
while (fscanf(fp, "%d", &n) == 1) {  
    printf("%d\n", n);  
}  
fclose(fp);  
return 0;
```

4) #include <stdio.h>

#include <stdlib.h>

```
int main(void) {  
    FILE *fp = fopen("texto.txt", "r");  
    if (!fp) { perror("Error ao abrir texto.txt");  
        return 1; }  
    char buf[510];  
    long linhas = 0;  
    while (fgets(buf, sizeof(buf), fp) != NULL) {  
        linhas++; }  
    fclose(fp);  
    printf("Número de linhas: %d\n", linhas);  
    return 0;
```

S T Q Q S S D

5) #include <stdio.h>

#include <ctype.h>

#include <stdlib.h>

int main(void) {

FILE *fp = fopen("texto.txt", "r");

If (!fp) { perror("Error ao abrir texto.txt"); }

return 1;

}

int c;

long palabras = 0;

int in_word = 0;

while ((c = fgetc(fp)) != EOF) {

If (!isspace(c)) {

If (!in_word) { palabras++; in_word = 1; }

else {

in_word = 0;

}

fclose(fp);

printf("Número de palabras: %d\n", palabras);

return 0;

}

6) #include <stdio.h>

int main() {

FILE *origem,

FILE *destino,

int c;

```
origem = fopen("origem.txt", "r");
if (origem == NULL) {
    printf("Erro ao abrir origem.txt!\n");
}
return 1;
```

```
destino = fopen("copia.txt", "w");
if (destino == NULL) {
    printf("Erro ao criar copia.txt!\n");
    fclose(origem);
}
return 1;
```

```
while ((c = fgetc(origem)) != EOF) {
    fputc(c, destino);
}
```

```
fclose(origem);
fclose(destino);
```

```
printf("Arquivo copiado com sucesso!\n");
return 0;
}
```

S T A A S S D

— / — / —

7) #include <stdio.h>

#include <stdlib.h>

int main () {

const int MAX = 40;

char nome[MAX];

int nota;

FILE *fs = fopen ("alunos.txt", "r");

if (fs == NULL)

exit (1);

while (fscanf (fs, "%s %d", nome, &nota) == 2) {

if (nota >= 7)

printf ("%s\n", nome);

}

fclose (fs);

fs = NULL

return 0;

}

8) #include <stdio.h>

#include <cctype.h>

#include <stdlib.h>

int main (void) {

FILE *fp = fopen ("texto.txt", "r");

if (!fp) { perror ("Erro ao abrir texto.txt");

enirab

S T Q Q S S D

} return 1;

long chars = 0, palavras = 0, linhas = 0;

int c;

int in_word = 0;

while ((c = fgetc(fp)) != EOF) {

chars++;

if (c == '\n') linhas++;

if (!isspace(c)) {

if (!in_word) { palavras++; in_word = 1; }

in_word = 0;

}

fclose(fp);

printf("Caracteres: %d\n", chars);

printf("Caracteres: %d\n Palavras: %d\n Linhas: %d\n", chars, palavras, linhas);

return 0;

}

9) #include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main(void) {

char palavras[128];

printf("Digite a palavra a pesquisar: ");

if (scanf("%127s", palavras) != 1)

spiral

S T Q Q S S D

— / — /

```

return 1; }

char linha [1024];
long cont = 0;
size_t plen = strlen(palavra);
while (fgets(linha, sizeof(linha), fp) != NULL) {
    char *p = linha;
    while ((p = strstr(p, palavra)) != NULL) {
        cont++;
        p += plen;
    }
    fclose(fp);
    printf("A palavra '%s' aparece %ld vezes.\n", palavra,
    cont);
}

return 0;
}

```

10) #include <stdio.h>
 #include <time.h>

```

int main(void) {
    FILE *f = fopen("log.txt", "a");
    if (!f) return 1;

    time_t agora = time(NULL);
    fprintf(f, "%s", ctime(&agora));
    fclose(f);
    return 0;
}

```

spiral

11) #include <stdio.h>
 #include <stdlib.h>

```
int main(void) {
    FILE *fp = fopen("texto.txt", "r");
    if (!fp) { perror("Erro ao abrir texto.txt");
    return 1; }
    char linha[1024];
    while (fscanf(fp, "%[^n]", linha) == 1) {
        printf("%s\n", linha);
    }
    fclose(fp);
}
return 0;
```

12) #include <stdio.h>
 #include <stdio.h>

```
int main(void) {
    FILE *fp = fopen("pessoas.txt", "r");
    if (!fp) { perror("Erro ao abrir pessoas.txt");
    return 1; }
    char nome[128];
    int pontuacao;
    int d, m, a;

    while (fscanf(fp, "%127[^,],%d,%d/%d/%d",
    nome, &pontuacao, &d, &m, &a) == 5) {
        printf("Nome: %s | Pontuacao: %d | Data:
        %02d/%02d/%04d\n", nome, pontuacao, d, m, a);
    }
}
```

spiral

S T Q Q S S D

— / — /

{

```
fclose(fp);
}
```

13) #include <stdio.h>
 #include <stdlib.h>

```
int main(void) {
```

```
FILE *fp = fopen("emails.txt", "r");
if (!fp) { perror("Erro ao abrir emails.txt"); return 1;
}
```

```
char usuario[128]; dominio[256];
```

```
while (fscanf(fp, "%127[^@]@%255s", usario,
  dominio) == 2) {
```

```
printf("Usario: %s | Dominio: %s\n", usario,
  dominio);
}
```

```
fclose(fp);
```

```
return 0;
}
```

14) #include <stdio.h>

#include <stdlib.h>

```
int cmp(const void *a, const void *b) {
```

```
return *(int *)a - *(int *)b;
}
```

```
int main(void) {
```

```
FILE *f = fopen("numeros.txt", "r");
```

S T Q Q S S D

```

if (!f) return 1;

int v[1000];
int n = 0;

while (fscanf(f, "%d", &v[n]) == 1);
    n++;

fclose(f);

qsort(v, n, sizeof(int), cmp);

f = fopen("ordenado.txt", "w");
if (!f) return 1;

for (int i = 0; i < n; i++)
    fprintf(f, "%d\n", v[i]);

fclose(f);

return 0;
}

```

15)

```

#include <stdio.h>
#include <stdlib.h>

```

```

typedef struct { char nome[100]; double nota; } Aluno;

```

```

int cmp(const void *a, const void *b) {
    double x = ((Aluno *)a) -> nota, y = ((Aluno *)b) -> nota;
    return (y > x) - (y < x);
}

```

spiral®

S T Q Q S S D

```
int main () {
    FILE *f = fopen ("alunos.txt", "r");
    if (!f) return 1;

    Aluno v[1000];
    int n = 0;

    while (fscanf (f, "%99s %lf", v[n].name, &v[n].nota) == 2)
        n++;
    fclose (f);

    qsort (v, n, sizeof (Aluno), cmp);

    f = fopen ("aprovados.txt", "w");
    if (!f) return 1;

    for (int i = 0; i < n; i++)
        fprintf (f, "%s %.2f\n", v[i].name, v[i].nota);
    fclose (f);
    return 0;
}
```