/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DAY 17: FILE HANDLING (BASIC) \*

\* Contains 10 C programs for basic file operations \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <ctype.h> // For toupper()

#include <string.h> // For strcmp()

// ==============================

// 1. Create File and Write String

// ==============================

void createAndWriteFile() {

FILE \*file = fopen("example.txt", "w");

if (file == NULL) {

printf("Error creating file!\n");

return;

}

fprintf(file, "This is a sample text written to file.\n");

fclose(file);

printf("File created and text written successfully.\n");

}

// ==============================

// 2. Read Contents from File

// ==============================

void readFileContents() {

FILE \*file = fopen("example.txt", "r");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

printf("File contents:\n");

char ch;

while ((ch = fgetc(file)) != EOF) {

putchar(ch);

}

fclose(file);

}

// ==============================

// 3. Copy File Contents

// ==============================

void copyFileContents() {

FILE \*source = fopen("example.txt", "r");

FILE \*dest = fopen("copy.txt", "w");

if (source == NULL || dest == NULL) {

printf("Error opening files!\n");

return;

}

char ch;

while ((ch = fgetc(source)) != EOF) {

fputc(ch, dest);

}

fclose(source);

fclose(dest);

printf("File copied successfully to copy.txt\n");

}

// ==============================

// 4. Count Characters in File

// ==============================

void countFileCharacters() {

FILE \*file = fopen("example.txt", "r");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

int count = 0;

while (fgetc(file) != EOF) {

count++;

}

fclose(file);

printf("Number of characters: %d\n", count);

}

// ==============================

// 5. Count Lines in File

// ==============================

void countFileLines() {

FILE \*file = fopen("example.txt", "r");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

int lines = 0;

char ch;

while ((ch = fgetc(file)) != EOF) {

if (ch == '\n') lines++;

}

fclose(file);

printf("Number of lines: %d\n", lines + 1); // +1 for last line without \n

}

// ==============================

// 6. Append to Existing File

// ==============================

void appendToFile() {

FILE \*file = fopen("example.txt", "a");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

fprintf(file, "This text is appended to the file.\n");

fclose(file);

printf("Text appended successfully.\n");

}

// ==============================

// 7. Display File Word by Word

// ==============================

void displayWords() {

FILE \*file = fopen("example.txt", "r");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

printf("File contents word by word:\n");

char word[100];

while (fscanf(file, "%s", word) != EOF) {

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

}

fclose(file);

}

// ==============================

// 8. Find Word Occurrence

// ==============================

void countWordOccurrence() {

FILE \*file = fopen("example.txt", "r");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

char search[50], word[50];

int count = 0;

printf("Enter word to search: ");

scanf("%s", search);

while (fscanf(file, "%s", word) != EOF) {

if (strcmp(word, search) == 0) {

count++;

}

}

fclose(file);

printf("'%s' occurs %d times.\n", search, count);

}

// ==============================

// 9. Remove Spaces from File

// ==============================

void removeSpacesFromFile() {

FILE \*source = fopen("example.txt", "r");

FILE \*temp = fopen("temp.txt", "w");

if (source == NULL || temp == NULL) {

printf("Error opening files!\n");

return;

}

char ch;

while ((ch = fgetc(source)) != EOF) {

if (ch != ' ') {

fputc(ch, temp);

}

}

fclose(source);

fclose(temp);

remove("example.txt");

rename("temp.txt", "example.txt");

printf("Spaces removed successfully.\n");

}

// ==============================

// 10. Convert File to Uppercase

// ==============================

void convertFileToUpper() {

FILE \*source = fopen("example.txt", "r");

FILE \*temp = fopen("temp.txt", "w");

if (source == NULL || temp == NULL) {

printf("Error opening files!\n");

return;

}

char ch;

while ((ch = fgetc(source)) != EOF) {

fputc(toupper(ch), temp);

}

fclose(source);

fclose(temp);

remove("example.txt");

rename("temp.txt", "example.txt");

printf("File converted to uppercase.\n");

}