1.

#include <stdio.h>

int main() {

FILE \*file;

char content[] = "Guess what? This file was made by a C program! \nIt's the digital equivalent of a high-five. \n";

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

if (file == NULL) {

printf("Oops! File couldn't be opened. Is your computer mad at me? 😅\n");

return 1;

}

fprintf(file, "%s", content);

fclose(file);

printf("Boom! File created and content written like a pro. 🎉\n");

return 0;

}

2.

#include <stdio.h>

int main() {

FILE \*file;

char ch;

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

if (file == NULL) {

printf("Whoops! Can't find the file. Did it run away? 😔\n");

return 1;

}

printf("Reading the file like a pro... 📖\n\n");

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

putchar(ch);

}

fclose(file);

printf("\nAll done! File reading complete. ✨\n");

return 0;

}

3.

#include <stdio.h>

int main() {

FILE \*file;

char ch;

int lines = 0;

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

if (file == NULL) {

printf("Uh-oh! No file found. Did it vanish into thin air? 👻\n");

return 1;

}

printf("Counting lines... prepare for a magical file journey! 🧙‍♂️\n");

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

if (ch == '\n') {

lines++;

}

}

fclose(file);

printf("We found %d line(s) in the file! 🎉\n", lines);

return 0;

}

4.

#include <stdio.h>

int main() {

FILE \*source, \*destination;

char ch;

source = fopen("source.txt", "r");

if (source == NULL) {

printf("Uh-oh! Can't find the source file. Did it teleport away? 🧐\n");

return 1;

}

destination = fopen("destination.txt", "w");

if (destination == NULL) {

printf("Oops! The destination file couldn't be created. Did the computer take a nap? 💤\n");

fclose(source);

return 1;

}

printf("Copying content... like a file ninja 🥷!\n");

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

fputc(ch, destination);

}

printf("Copy complete! Your data is safely transported. 🛸\n");

fclose(source);

fclose(destination);

return 0;

}

5.

#include <stdio.h>

int main() {

FILE \*file;

char text[] = "This is the new line being added to the file! 🌟\n";

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

if (file == NULL) {

printf("Oops! The file is missing or is taking a nap. 😴\n");

return 1;

}

printf("Appending new text... Let the magic begin! ✨\n");

fprintf(file, "%s", text);

printf("Text added! File is now even more awesome! 🎉\n");

fclose(file);

return 0;

}

6.

#include <stdio.h>

int main() {

FILE \*file;

char ch;

int vowelCount = 0;

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

if (file == NULL) {

printf("Uh-oh! Can't find the file. Is it off to a secret lair? 🕵️‍♂️\n");

return 1;

}

printf("Counting vowels... Here we go! 🧠💨\n");

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

ch = tolower(ch); // Make the character lowercase to count both uppercase and lowercase vowels

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

vowelCount++;

}

}

fclose(file);

printf("Found %d vowels in the file. Vowel power! 💪\n", vowelCount);

return 0;

}

7.

#include <stdio.h>

int main() {

FILE \*file;

int num, sum = 0;

file = fopen("numbers.txt", "r");

if (file == NULL) {

printf("Oops! The file is missing. Did it go on vacation? 🌴\n");

return 1;

}

printf("Reading numbers... Time to summon some math powers! 🧙‍♂️\n");

while (fscanf(file, "%d", &num) == 1) {

sum += num;

}

fclose(file);

printf("The sum of all the numbers in the file is: %d 🥳\n", sum);

return 0;

}

8.

#include <stdio.h>

struct Person {

char name[50];

int age;

};

int main() {

FILE \*file;

struct Person person;

// Open the file for reading

file = fopen("person.txt", "r");

if (file == NULL) {

printf("Uh-oh! Can't find the file. Did it run off somewhere? 🕵️‍♂️\n");

return 1;

}

// Read the structure from the file

fread(&person, sizeof(struct Person), 1, file);

fclose(file);

// Display the contents of the structure

printf("Person details from the file:\n");

printf("Name: %s\n", person.name);

printf("Age: %d\n", person.age);

return 0;

}

9.

#include <stdio.h>

#include <string.h>

#define MAX\_NAMES 100

#define MAX\_LENGTH 50

int main() {

FILE \*file;

char names[MAX\_NAMES][MAX\_LENGTH];

int count = 0;

char temp[MAX\_LENGTH];

file = fopen("names.txt", "r");

if (file == NULL) {

printf("Uh-oh! Can't find the file. Did it run away? 😅\n");

return 1;

}

printf("Reading names from the file... Let the sorting begin! 🎉\n");

while (fgets(names[count], MAX\_LENGTH, file) != NULL) {

names[count][strcspn(names[count], "\n")] = 0; // Remove newline character

count++;

}

fclose(file);

// Sorting the names alphabetically

for (int i = 0; i < count - 1; i++) {

for (int j = i + 1; j < count; j++) {

if (strcmp(names[i], names[j]) > 0) {

// Swap the names

strcpy(temp, names[i]);

strcpy(names[i], names[j]);

strcpy(names[j], temp);

}

}

}

10.

#include <stdio.h>

#include <string.h>

#define MAX\_LENGTH 100

int main() {

FILE \*file;

char line[MAX\_LENGTH];

char word[MAX\_LENGTH];

int found = 0;

// Ask for the word to search

printf("Enter the word to search for: ");

scanf("%s", word);

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

if (file == NULL) {

printf("Oops! Can't find the file. Is it off on a secret mission? 🤔\n");

return 1;

}

printf("Searching for the word '%s'... 🔍\n", word);

// Read the file line by line and search for the word

while (fgets(line, MAX\_LENGTH, file)) {

if (strstr(line, word) != NULL) {

printf("Found it! The word '%s' is here: %s", word, line);

found = 1;

break;

}

}

if (!found) {

printf("Hmm... couldn't find the word '%s'. Maybe it's hiding? 😅\n", word);

}

fclose(file);

return 0;

}