// 1. Write a string N times in a file

#include <stdio.h>

int main() {

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

if (!fp) return 1;

char str[] = "Hello World\n";

int n = 5;

for (int i = 0; i < n; i++) {

fputs(str, fp);

}

fclose(fp);

printf("String written %d times.\n", n);

return 0;

}

// 2. Count the number of digits in a file

#include <stdio.h>

#include <ctype.h>

int main() {

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

if (!fp) return 1;

int digits = 0;

int ch;

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

if (isdigit(ch))

digits++;

}

fclose(fp);

printf("Number of digits in file: %d\n", digits);

return 0;

}

// 3. Find and replace a word in a file (simple version)

#include <stdio.h>

#include <string.h>

int main() {

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

if (!fp) return 1;

FILE \*fp2 = fopen("temp.txt", "w");

if (!fp2) { fclose(fp); return 1; }

char line[256];

char find[] = "old";

char replace[] = "new";

while (fgets(line, sizeof(line), fp)) {

char \*pos = strstr(line, find);

if (pos) {

\*pos = '\0';

fprintf(fp2, "%s%s%s", line, replace, pos + strlen(find));

} else {

fputs(line, fp2);

}

}

fclose(fp);

fclose(fp2);

remove("file.txt");

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

printf("Word replaced in file.\n");

return 0;

}

// 4. Print only lines containing a specific word

#include <stdio.h>

#include <string.h>

int main() {

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

if (!fp) return 1;

char word[] = "hello";

char line[256];

printf("Lines containing '%s':\n", word);

while (fgets(line, sizeof(line), fp)) {

if (strstr(line, word)) {

printf("%s", line);

}

}

fclose(fp);

return 0;

}

// 5. Merge contents of two files into a third

#include <stdio.h>

void copy\_file(FILE \*src, FILE \*dest) {

int ch;

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

fputc(ch, dest);

}

}

int main() {

FILE \*f1 = fopen("file1.txt", "r");

FILE \*f2 = fopen("file2.txt", "r");

FILE \*f3 = fopen("file3.txt", "w");

if (!f1 || !f2 || !f3) return 1;

copy\_file(f1, f3);

copy\_file(f2, f3);

fclose(f1);

fclose(f2);

fclose(f3);

printf("Files merged into file3.txt\n");

return 0;

}

// 6. Reverse the contents of a file

#include <stdio.h>

#include <stdlib.h>

int main() {

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

if (!fp) return 1;

fseek(fp, 0, SEEK\_END);

long size = ftell(fp);

char \*buffer = malloc(size);

if (!buffer) { fclose(fp); return 1; }

fseek(fp, 0, SEEK\_SET);

fread(buffer, 1, size, fp);

fclose(fp);

FILE \*fp2 = fopen("reversed.txt", "w");

if (!fp2) { free(buffer); return 1; }

for (long i = size - 1; i >= 0; i--) {

fputc(buffer[i], fp2);

}

fclose(fp2);

free(buffer);

printf("File reversed.\n");

return 0;

}

// 7. Print only even-numbered lines from a file

#include <stdio.h>

int main() {

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

if (!fp) return 1;

char line[256];

int lineNum = 1;

printf("Even-numbered lines:\n");

while (fgets(line, sizeof(line), fp)) {

if (lineNum % 2 == 0)

printf("%s", line);

lineNum++;

}

fclose(fp);

return 0;

}

// 8. Print characters between two given positions

#include <stdio.h>

int main() {

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

if (!fp) return 1;

int start = 5, end = 15;

int pos = 1, ch;

printf("Characters between positions %d and %d:\n", start, end);

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

if (pos >= start && pos <= end)

putchar(ch);

if (pos > end)

break;

pos++;

}

fclose(fp);

return 0;

}

// 9. Check whether a file exists or not

#include <stdio.h>

int main() {

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

if (fp) {

printf("File exists.\n");

fclose(fp);

} else {

printf("File does not exist.\n");

}

return 0;

}

// 10. Copy only capital letters from one file to another

#include <stdio.h>

#include <ctype.h>

int main() {

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

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

if (!src || !dest) return 1;

int ch;

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

if (isupper(ch))

fputc(ch, dest);

}

fclose(src);

fclose(dest);

printf("Capital letters copied to capitals.txt\n");

return 0;

}