FILE

1. Write a C Program to list all files and sub-directories in a directory.

#include <stdio.h>

#include <dirent.h>

int main() {

DIR \*directory;

struct dirent \*entry;

char path[100];

printf("Enter the path of the directory: ");

gets(path);

directory = opendir(path);

if (directory == NULL) {

printf("Error opening directory.\n");

return 1;

}

printf("\nFiles and sub-directories in '%s':\n", path);

while ((entry = readdir(directory)) != NULL) {

printf("%s\n", entry->d\_name);

}

closedir(directory);

return 0;

}

1. Write a C Program to count number of lines in a file.

#include <stdio.h>

int main() {

FILE \*file;

char filename[50], ch;

int lines = 0;

printf("Enter the name of the file: ");

gets(filename);

file = fopen(filename, "r");

if (file == NULL) {

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

return 1;

}

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

if (ch == '\n') {

lines++;

}

}

fclose(file);

printf("Number of lines in '%s': %d\n", filename, lines);

return 0;

}

1. Write a C Program to print contents of file.

#include <stdio.h>

int main() {

FILE \*file;

char filename[50], ch;

printf("Enter the name of the file: ");

gets(filename);

file = fopen(filename, "r");

if (file == NULL) {

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

return 1;

}

printf("\nContents of '%s':\n", filename);

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

printf("%c", ch);

}

fclose(file);

return 0;

}

1. Write a C Program to copy contents of one file to another file.

#include <stdio.h>

int main() {

FILE \*sourceFile, \*destinationFile;

char sourceFilename[50], destinationFilename[50], ch;

printf("Enter the name of the source file: ");

gets(sourceFilename);

sourceFile = fopen(sourceFilename, "r");

if (sourceFile == NULL) {

printf("Error opening source file.\n");

return 1;

}

printf("Enter the name of the destination file: ");

gets(destinationFilename);

destinationFile = fopen(destinationFilename, "w");

if (destinationFile == NULL) {

fclose(sourceFile);

printf("Error opening destination file.\n");

return 1;

}

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

fputc(ch, destinationFile);

}

printf("Contents copied successfully from '%s' to '%s'.\n", sourceFilename, destinationFilename);

fclose(sourceFile);

fclose(destinationFile);

return 0;

}

1. Write a C Program to merge contents of two files into a third file.

#include <stdio.h>

int main() {

FILE \*file1, \*file2, \*mergedFile;

char file1Name[50], file2Name[50], mergedFileName[50], ch;

printf("Enter the name of the first file: ");

gets(file1Name);

file1 = fopen(file1Name, "r");

if (file1 == NULL) {

printf("Error opening first file.\n");

return 1;

}

printf("Enter the name of the second file: ");

gets(file2Name);

file2 = fopen(file2Name, "r");

if (file2 == NULL) {

fclose(file1);

printf("Error opening second file.\n");

return 1;

}

printf("Enter the name of the merged file: ");

gets(mergedFileName);

mergedFile = fopen(mergedFileName, "w");

if (mergedFile == NULL) {

fclose(file1);

fclose(file2);

printf("Error opening merged file.\n");

return 1;

}

// Copy contents of the first file to the merged file

while ((ch = fgetc(file1)) != EOF) {

fputc(ch, mergedFile);

}

// Copy contents of the second file to the merged file

while ((ch = fgetc(file2)) != EOF) {

fputc(ch, mergedFile);

}

printf("Contents of '%s' and '%s' merged successfully into '%s'.\n", file1Name, file2Name, mergedFileName);

fclose(file1);

fclose(file2);

fclose(mergedFile);

return 0;

}

1. Write a C program to delete a file.

#include <stdio.h>

int main() {

char filename[50];

printf("Enter the name of the file to be deleted: ");

gets(filename);

if (remove(filename) == 0) {

printf("File '%s' deleted successfully.\n", filename);

} else {

printf("Error deleting file '%s'.\n", filename);

}

return 0;

}