Files in C

1. Introduction to File Handling in C

1. **Overview of File Handling Concepts:**
    - A file is a collection of related data stored on a disk.
    - File handling involves operations such as reading from and *writing to files*[7 p.36].
2. **File Operations:**

```
#include <stdio.h>
int main() {
FILE *file_ptr;
file_ptr = fopen("example.txt", "r");
if (file_ptr == NULL)
{ printf("Error opening file!\n");
return 1; } // File operations
fclose(file_ptr);
return 0; }
```

3. **Understanding File Streams:**
    - `FILE` is a data structure defined in `<stdio.h>` to handle files.
    - `stdin`, `stdout`, and `stderr` are predefined file pointers.
4. **Error Handling and File I/O Errors:**
    - Error handling during file operations:

```
FILE *file_ptr = fopen("example.txt", "r");
if (file_ptr == NULL)
{ perror("Error opening file");
return 1; }
```

2. Reading from Files

1. **Sequential File Access:**

```
char ch;
while ((ch = fgetc(file_ptr)) != EOF)
{ // Process character }
```

2. **Reading Character-by-Character:**

```
char ch = fgetc(file_ptr);
```

3. **Reading Line-by-Line:**

```
char buffer[100];
while (fgets(buffer, sizeof(buffer), file_ptr) != NULL)
{ // Process line }
```

4. **Using Formatted Input Functions:**

```
int num;
fscanf(file_ptr, "%d", &num);
```

3. Writing to Files

1. **Sequential File Output:**

```
fprintf(file_ptr, "Hello, world!\n");
```

2. **Writing Character-by-Character:**

```
char ch = 'A';
fputc(ch, file_ptr);
```

3. **Writing Line-by-Line:**

```
fputs("This is a line.", file_ptr);
```

4. **Using Formatted Output Functions:**

```
int num = 10;
fprintf(file_ptr, "%d", num);
```

4. Practical Projects

1. **Simple Text Editor:**
    - Implement functions for opening, editing, saving, and closing text files.
2. **File Encryption Program:**
    - Develop functions for encrypting and decrypting files using chosen algorithms.
3. **Database Management System:**
    - Create functions for storing and retrieving data from files in a structured format.

Exercise solutions

Solution n°1

[exercice p. 6]

What does `main()` represent in C?

In C, main() is a function

Solution n°2

[exercice p. 6]

What do we mean by `main()` function

- ☑ In C programming, the main() function is an essential component of every C program.Typically, when the C program is running, the operating system calls the function to initiate the execution of the program's code.
- ☑ The main function in C **marks the beginning of any program in C**. The main function in C is the first function to be executed by the Operating System.

Solution n°3

[exercice p. 7]

why we use curly braces {} in C ?

- ☑ The curly braces **denote a block of code, in which variables can be declared**.
- ☑ {and} are used to limit the scope of declarations and to act as a single statement for control structures.

Solution n°4

[exercice p. 13]

What is an array in C?

- ☐ A single variable that can hold multiple values of different data types
- ☑ A collection of elements of the same data type stored under a single identifier
- ☐ A reserved keyword used to define functions

Solution n°5

[exercice p. 13]

How do you declare an array in C?

- ☑ int array[5];
- ☐ array = int[5];

Solution n°6

[exercice p. 14]

What is the index of the first element in an array in C?

- [x] 0
- [ ] 1

Solution n°7

[exercice p. 14]

How do you access the third element of an array named `numbers` in C?

- [ ] numbers(3);
- [ ] numbers{2};
- [x] numbers[2];