FILE HANDLING C

- to handle files for performing operations like reading, writing, and modifying files.

1. **fopen():**

   – **Syntax:**

   ```
   FILE *fopen(const char *filename, const char *mode);
   ```

   Return Value:

   - Success: Pointer to the FILE object.
   - Failure: NULL.

   Usage:

   – Opens a file in a specified mode.

   Modes:

   - "r": Read (file must exist).
   - "w": Write (creates/overwrites).
   - "a": Append (creates if not exists).
   - "r+": Read/Write.
   - "w+": Write/Read (overwrites).
   - "a+": Append/Read.

2. **fclose():**

   – **Syntax:**

   ```
   fclose(FILE *stream);
   ```

   Return Value:

   - Success: 0.
   - Failure: Non-zero.

   Usage: Closes an open file.

3. **fprintf():**

   – **Syntax:**

   ```
   int fprintf(FILE *stream, const char *format, ...);
   ```

   Return Value:

   - Success: Number of characters written.

- Failure: Negative value.

Usage: Writes formatted text to a file.

4. **fscanf():**

– **Syntax:**

```
int fscanf(FILE *stream, const char *format, ...);
```

Return Value:

- Success: Number of items successfully read.
- Failure: EOF.

Usage: Reads formatted input from a file.

5. **fgetc():**

– **Syntax:**

```
int fgetc(FILE *stream);
```

Return Value:

- Success: Character read (as int).
- Failure: EOF.

Usage: Reads a single character from a file.

6. **fputc():**

– **Syntax:**

```
int fputc(int char, FILE *stream);
```

Return Value:

- Success: Character written.
- Failure: EOF.

Usage: Writes a single character to a file.

7. **fgets():**

– **Syntax:**

```
char *fgets(char *str, int n, FILE *stream);
```

Return Value:

- Success: Pointer to the string.
- Failure: NULL.

Usage: Reads a line from a file.

8. **fputs():**

– **Syntax:**

int fputs(const char *str, FILE *stream);

Return Value:

- Success: Non-negative value.
- Failure: EOF.

Usage: Writes a string to a file.

9. fread():

– **Syntax:**

size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);

Return Value: Number of elements read.

Usage: Reads binary data from a file.

10. fwrite():

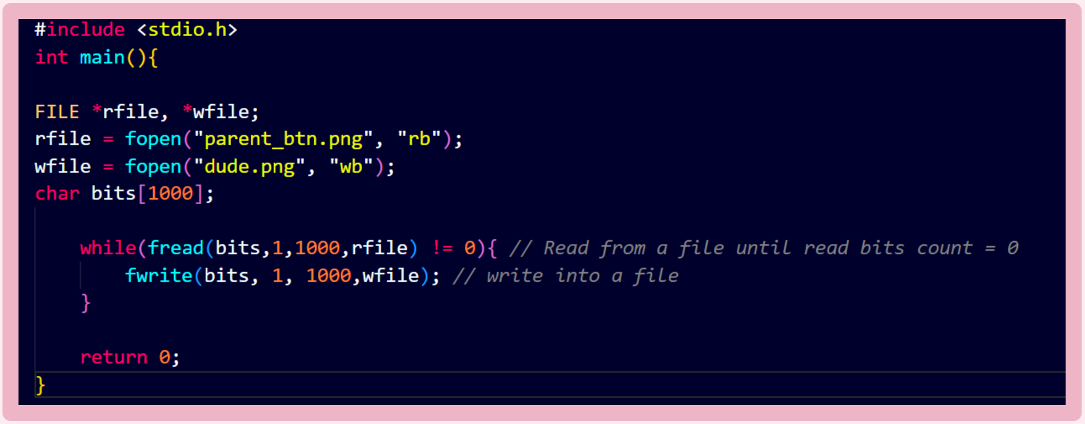
– **Syntax:**

size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);

Return Value: Number of elements written.

Usage: Writes binary data to a file.

Example:



```c
#include <stdio.h>
int main(){

FILE *rfile, *wfile;
rfile = fopen("parent_btn.png", "rb");
wfile = fopen("dude.png", "wb");
char bits[1000];

    while(fread(bits,1,1000,rfile) != 0){ // Read from a file until read bits count = 0
        fwrite(bits, 1, 1000,wfile); // write into a file
    }

    return 0;
}
```

11. fseek():

– **Syntax:**

int fseek(FILE *stream, long offset, int whence);

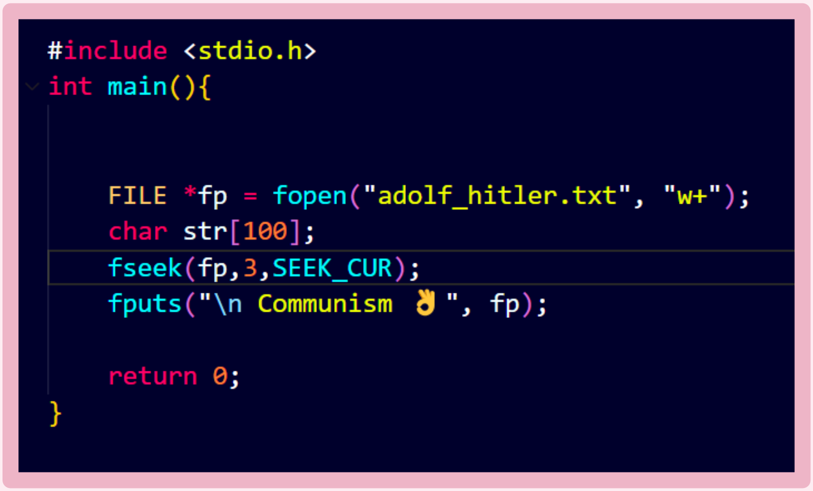
Return Value:

- Success: 0.
- Failure: Non-zero.

Usage: Moves the file pointer to a specific position.
Modes:

- SEEK_SET: Beginning of the file.
- SEEK_CUR: Current position.
- SEEK_END: End of the file.

Example:



```
#include <stdio.h>
int main(){


    FILE *fp = fopen("adolf_hitler.txt", "w+");
    char str[100];
    fseek(fp,3,SEEK_CUR);
    fputs("\n Communism 👌 ", fp);

    return 0;
}
```

12. ftell():

– **Syntax:**

long ftell(FILE *stream);

Return Value:

- Success: Current file pointer position.
- Failure: -1.

Usage: Returns the current position in a file.

13. rewind():

- **Syntax:**

void rewind(FILE *stream);

Return Value: None.

Usage: Moves the file pointer to the beginning.

14. remove():

- **Syntax:**

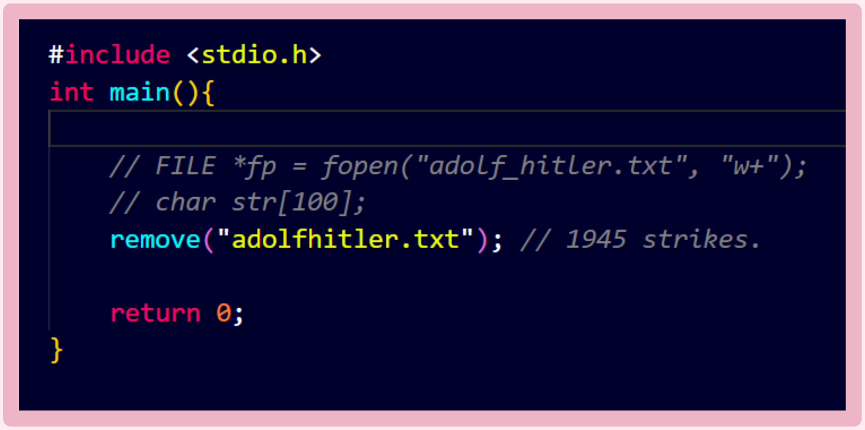
int remove(const char *filename);

Return Value:

- Success: 0.
- Failure: Non-zero.

Usage: Deletes a file.

Example:



```c
#include <stdio.h>
int main(){

    // FILE *fp = fopen("adolf_hitler.txt", "w+");
    // char str[100];
    remove("adolfhitler.txt"); // 1945 strikes.

    return 0;
}
```

15. rename():

- **Syntax:**

int rename(const char *oldname, const char *newname);

Return Value:

- Success: 0.
- Failure: Non-zero.

Usage: Renames or moves a file.

To Move File:
rename("file.txt", "new_directory/file.txt") ← Pass directory name in new name.

16. getw() and putw() (Obsolete):

– **Syntax:**

int getw(FILE *stream);

int putw(int w, FILE *stream);

Return Value:

- getw: Returns the next integer from the file.
- putw: Returns the written integer on success; EOF on failure.

Usage:
Reads/writes integers to/from a binary file. **These are largely replaced by fread and fwrite.**