Andrea Fanti 235808

Lab 2 exercise 1

The Makefile should call the C compiler with the -g option to not perform code optimizations and to produce debugging information, that will be later used by `gdb`.

After compiling with `make`, the program can be run with `./test-gdb`
The program executes and creates a file, `Out.txt.v1`, that is different from `In.txt`, as we can see with `diff`.

To have more information about the differences we can use the command `od -ab <file>`, that prints the content of the file in octal form along its ASCII decoding.
After running `od` on both files, we can see that `Out.txt.v1` is a copy of the input file with an additional byte (of octal value 377: all ones) appended at the end.

To perform the requested operations with `gdb`, we have to write these commands:

1. b 15
2. r
3. p name
4. n
5. p name
6. s
7. b 26
8. c
9. p fpR
10. p *fpR
11. b 32
12. (continue until we understand the problem)

The problem with this program is that when we reach the end of the file with the last "useful" read, the EOF flag is not set, so the `while` condition is still true, and the read/write is performed another time.
The `fgetc(fpR)` instruction returns the `EOF` constant, that has usually value -1 (that converted in binary is all ones). This value is then truncated from 32 bits (usually) to 8 bits (a character dimension) and written to the output file.

The program can be modified moving the `fgetc` line into the while condition.
The final result for the while loop is

```
while ( (c = fgetc (fpR)) != EOF){
      fputc (c, fpW);
   }
```

and this works without problems.