

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include "assignment1.h"

int main()
{
    char byte[9];
    FILE *f = fopen("input.txt", "r"); //open input.txt
    int c;
    signed char *line = NULL;
    size_t length = 0;
    while (c = fgetc(f), c != EOF) { //read character in file f until EOF
        ungetc(c, f); //push string back to f
        while (c = fgetc(f), c != EOF && c != '\n') { //read until EOF or new line
            ungetc(c, f); ///push string back to f
            length += 8; //increase size by 8 bits
            line = realloc(line, length); //adjust the allocation of line to be length size
            for (size_t i = length - 8; i < length; i++) {
                fgets(byte, 9, f); //read 8 characters string (and null terminator) from f into array byte
                line[i] = strtol(byte, NULL, 2); //convert byte to long integer base 2, store in signed char
            }
        }
    }

    fclose(f); //close input.txt
    for (size_t i = 0; i < length/2; i++) { //reverse byte order for little endian
        signed char temp = line[i];
        line[i] = line[length - i - 1];
        line[length - i - 1] = temp;
    }
    f = fopen("output.txt", "w"); //create output.txt and write into it

    //write in output.txt signed char
    fprintf(f, "Signed Char: ");
    for (size_t i = 0; i < length; i++) {
        fprintf(f, "%hhhd ", line[i]);
    }

    //write in output.txt ASCII codes
    fprintf(f, "\nASCII Codes: ");
    for (size_t i = 0; i < length; i++) {
        fprintf(f, "%c ", isprint(line[i]) ? line[i] : '.');
    }

    //write in output.txt unsigned char
    fprintf(f, "\nUnsigned Char: ");
    for (size_t i = 0; i < length; i++) {
        fprintf(f, "%hhu ", line[i]);
    }

    //write in output.txt signed int

```

```
fprintf(f, "\nSigned Int: ");
for (size_t i = 0; i < length; i+=4) {
    fprintf(f, "%d ", *(signed int *) (line+i));
}

//write in output.txt unsigned int
fprintf(f, "\nUnsigned Int: ");
for (size_t i = 0; i < length; i+=4) {
    fprintf(f, "%u ", *(unsigned int *) (line+i));
}

//write in output.txt signed float
fprintf(f, "\nSigned Float: ");
for (size_t i = 0; i < length; i+=4) {
    fprintf(f, "%0.4f ", *(float *) (line+i));
}

//write in output.txt signed double
fprintf(f, "\nSigned Double: ");
for (size_t i = 0; i < length; i+=8) {
    fprintf(f, "%0.4f ", *(double *) (line+i));
}

free(line); //free memory allocated by line
fclose(f); //close input.txt
return 0;
}
```