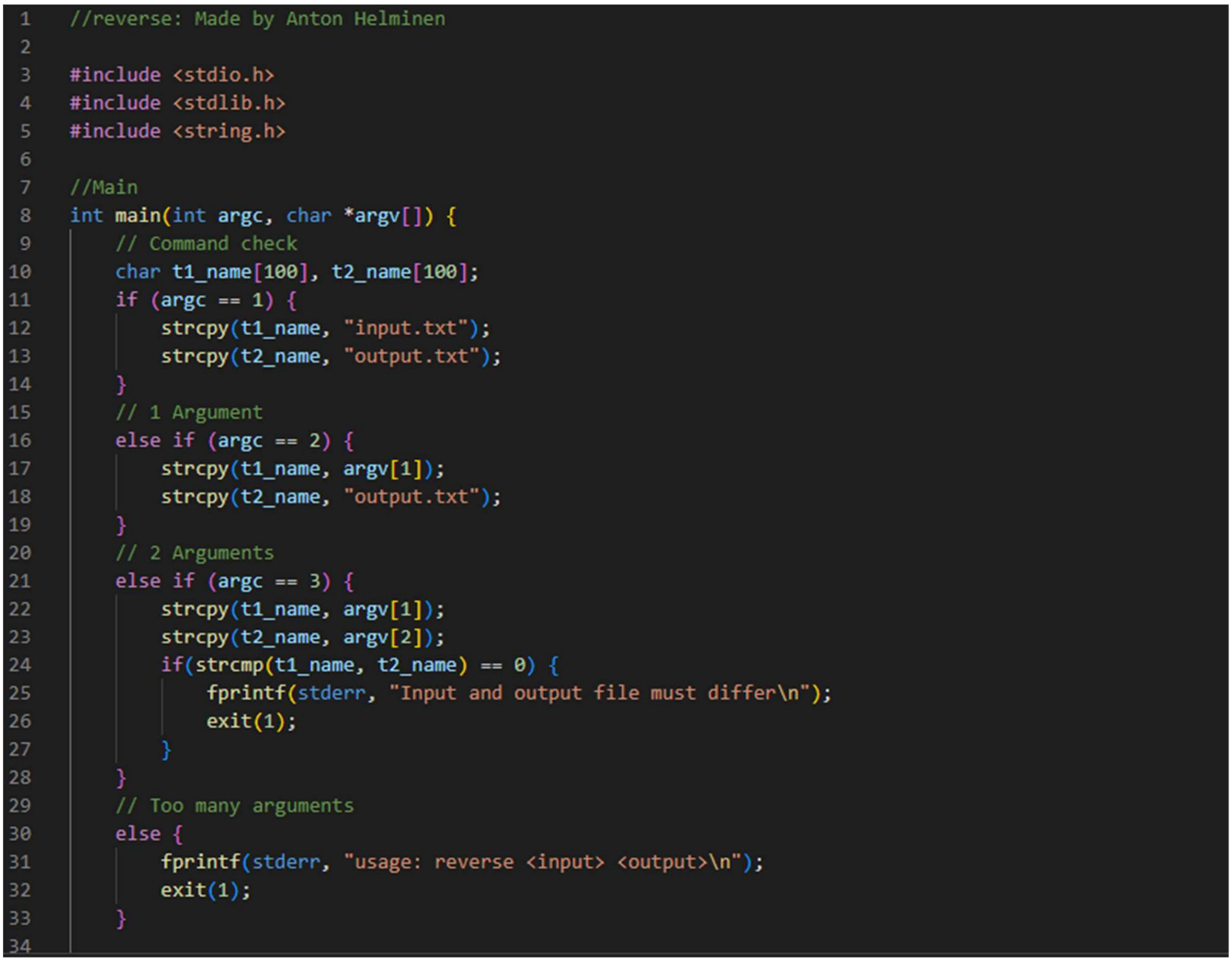
Project 1: Warmup to C and Unix programming

My solution reads an input text file and produces the same text file in reverse. This reversing is done by combining strings and reversing their characters and the characters' positions. This results in a reverse string which is then printed to a new file. The code works well for smaller files, but I failed to create a dynamic solution for memory allocation. Therefore, not all assumptions are addressed. Errors have been addressed, however.

The code consists of only one function: main() and it handles all processes in a linear fashion.

Screenshots of program, source code is also found in a separate file.

First half



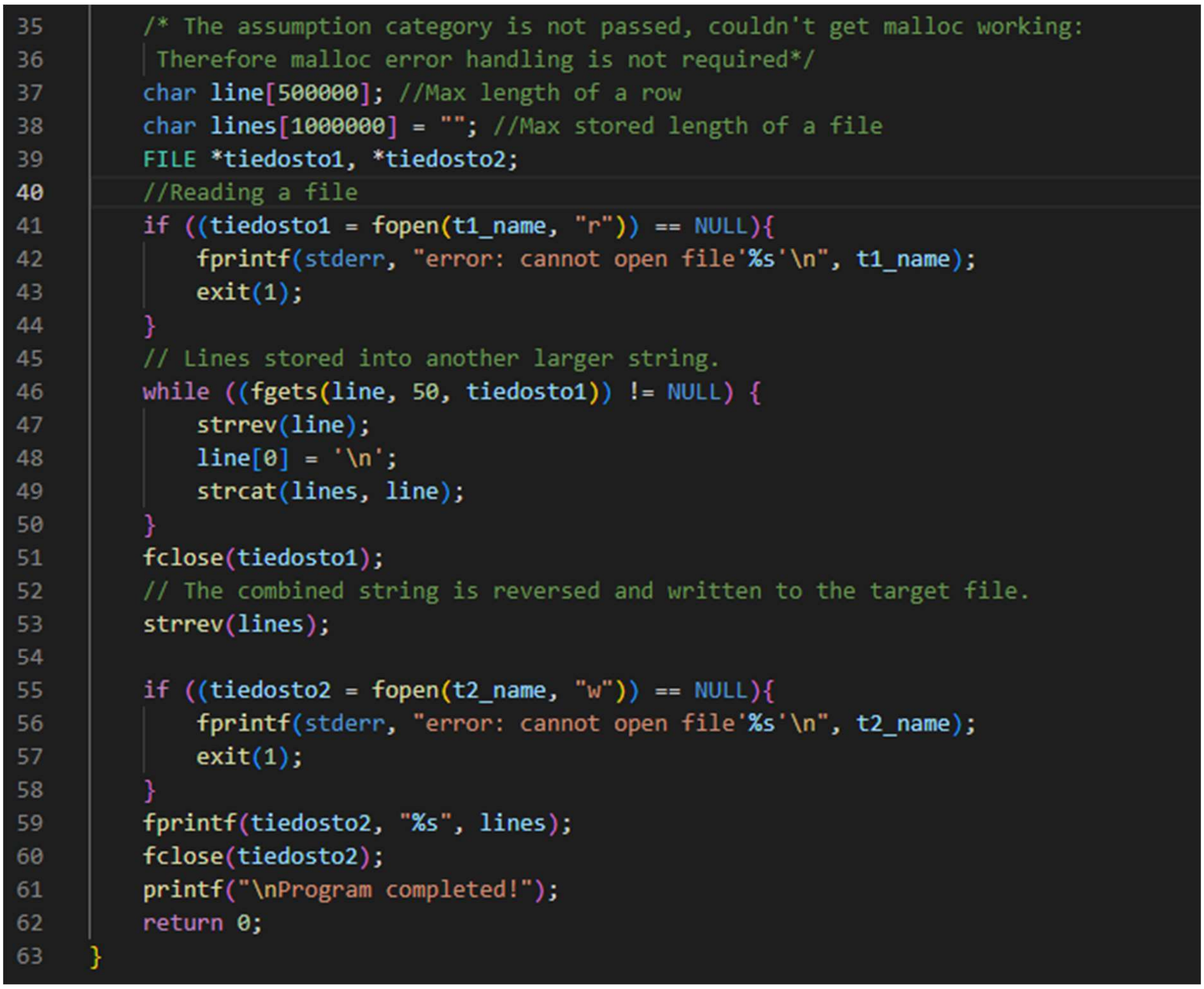
```c
//reverse: Made by Anton Helminen

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

//Main
int main(int argc, char *argv[]) {
    // Command check
    char t1_name[100], t2_name[100];
    if (argc == 1) {
        strcpy(t1_name, "input.txt");
        strcpy(t2_name, "output.txt");
    }
    // 1 Argument
    else if (argc == 2) {
        strcpy(t1_name, argv[1]);
        strcpy(t2_name, "output.txt");
    }
    // 2 Arguments
    else if (argc == 3) {
        strcpy(t1_name, argv[1]);
        strcpy(t2_name, argv[2]);
        if(strcmp(t1_name, t2_name) == 0) {
            fprintf(stderr, "Input and output file must differ\n");
            exit(1);
        }
    }
    // Too many arguments
    else {
        fprintf(stderr, "usage: reverse <input> <output>\n");
        exit(1);
    }

```

Second half

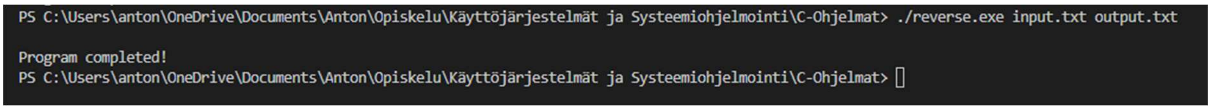


```c
    /* The assumption category is not passed, couldn't get malloc working:
     Therefore malloc error handling is not required*/
    char line[500000]; //Max length of a row
    char lines[1000000] = ""; //Max stored length of a file
    FILE *tiedosto1, *tiedosto2;
    //Reading a file
    if ((tiedosto1 = fopen(t1_name, "r")) == NULL){
        fprintf(stderr, "error: cannot open file'%s'\n", t1_name);
        exit(1);
    }
    // Lines stored into another larger string.
    while ((fgets(line, 50, tiedosto1)) != NULL) {
        strrev(line);
        line[0] = '\n';
        strcat(lines, line);
    }
    fclose(tiedosto1);
    // The combined string is reversed and written to the target file.
    strrev(lines);

    if ((tiedosto2 = fopen(t2_name, "w")) == NULL){
        fprintf(stderr, "error: cannot open file'%s'\n", t2_name);
        exit(1);
    }
    fprintf(tiedosto2, "%s", lines);
    fclose(tiedosto2);
    printf("\nProgram completed!");
    return 0;
}
```

Run examples:

Run example with 2 parameters (Both valid)

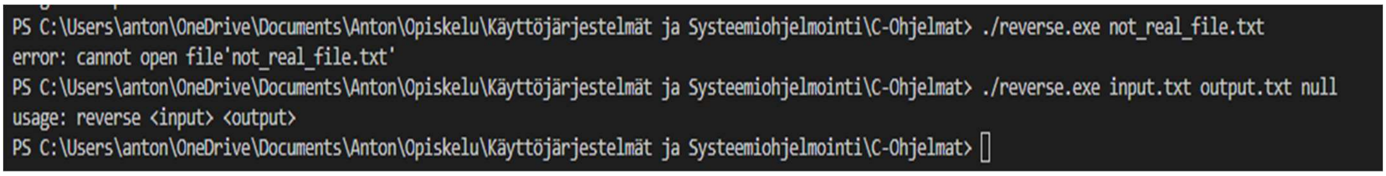


```
PS C:\Users\anton\OneDrive\Documents\Anton\Opiskelu\Käyttöjärjestelmät ja Systeemiohjelmointi\C-Ohjelmat> ./reverse.exe input.txt output.txt

Program completed!
PS C:\Users\anton\OneDrive\Documents\Anton\Opiskelu\Käyttöjärjestelmät ja Systeemiohjelmointi\C-Ohjelmat> 
```

Run example with incorrect input file and

Run example with too many parameters



```
PS C:\Users\anton\OneDrive\Documents\Anton\Opiskelu\Käyttöjärjestelmät ja Systeemiohjelmointi\C-Ohjelmat> ./reverse.exe not_real_file.txt
error: cannot open file'not_real_file.txt'
PS C:\Users\anton\OneDrive\Documents\Anton\Opiskelu\Käyttöjärjestelmät ja Systeemiohjelmointi\C-Ohjelmat> ./reverse.exe input.txt output.txt null
usage: reverse <input> <output>
PS C:\Users\anton\OneDrive\Documents\Anton\Opiskelu\Käyttöjärjestelmät ja Systeemiohjelmointi\C-Ohjelmat> 
```