Progressive Report on Week 1 Assignment Part B: helloArgv

Student Name Alain NIGANZE
Email: niganzealain@gmail.com
Tel:0783943932
Date:15/06/2024

Introduction

The objective of this assignment was to extend a simple "Hello, World!" program in C to handle command-line arguments and perform different actions based on those arguments. The enhanced program, named `helloArgv`, should print welcome messages according to specified command-line options and display a help message when needed.

Task Breakdown

1. Setting Up the Environment

- I created a new directory for the assignment using the command:

mkdir Week1B

- Then, I navigated into the directory:

cd Week1B

2. Creating and Running `helloworld.c`

- I created the initial `helloworld.c` file using:

touch helloworld.c

- I edited `helloworld.c` to include a basic "Hello, World!" program.
- I compiled and ran the program:

gcc -o helloworld helloworld.c
./helloworld

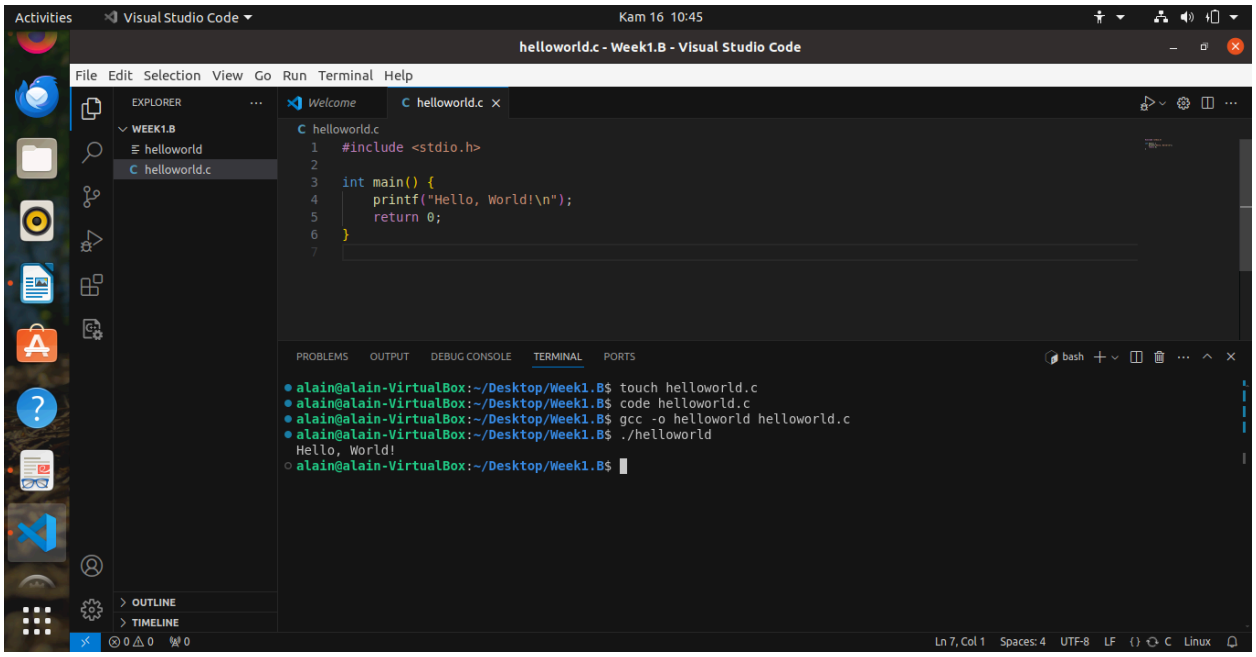


Figure1: Terminal showing compilation and running of `helloworld.c`

3. Renaming and Extending the File

- I renamed `helloworld.c` to `helloArgv.c` using the command:

mv helloworld.c helloArgv.c

- I edited `helloArgv.c` to handle command-line arguments and perform actions based on those arguments.

4. Implementing Command-Line Parameter Handling

- The program was modified to:
- Print a morning welcome message with the `-m` option.
- Print an evening welcome message with the `-e` option.
- Print multiple morning or evening messages when followed by a number (e.g., `-m 5` for five morning messages).

- Provide a help message with the `--help` option.

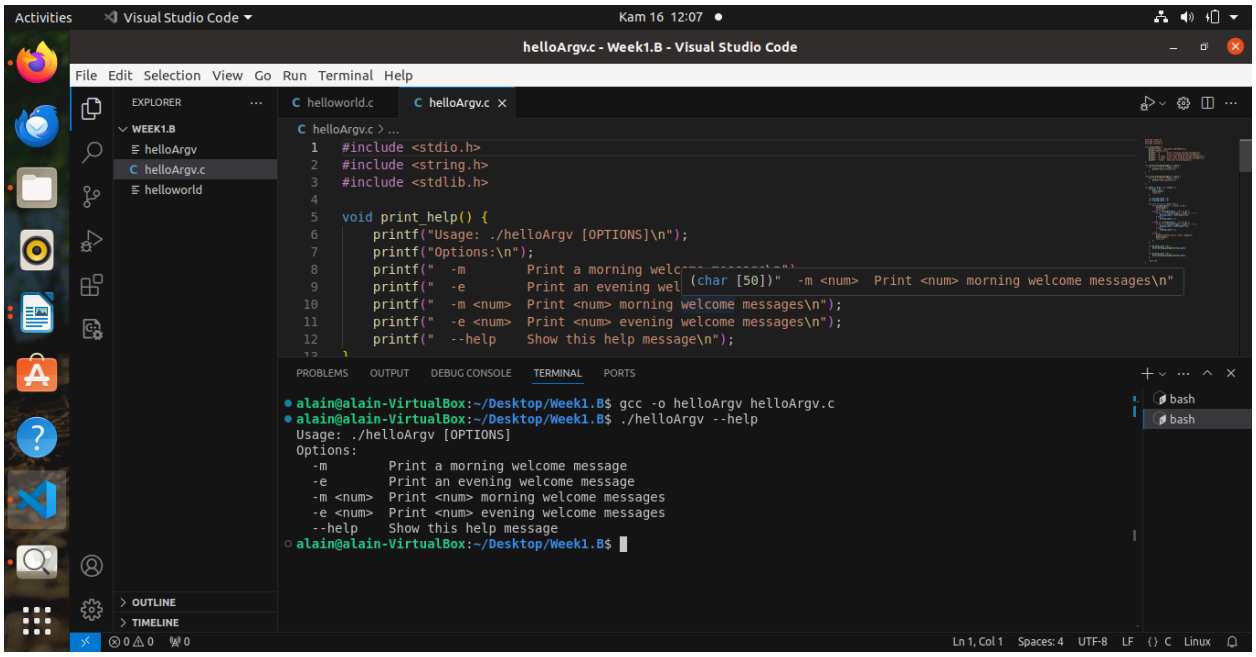


Figure2: show terminal compilation when we run -- help.

5. Compiling and Running `helloArgv.c'

- I compiled the extended program using:

```sh
gcc -o helloArgv helloArgv.c
```

- I tested the program with various command-line inputs:

```sh
./helloArgv -m
./helloArgv -e
./helloArgv -m 5 -e 3
./helloArgv --help
```

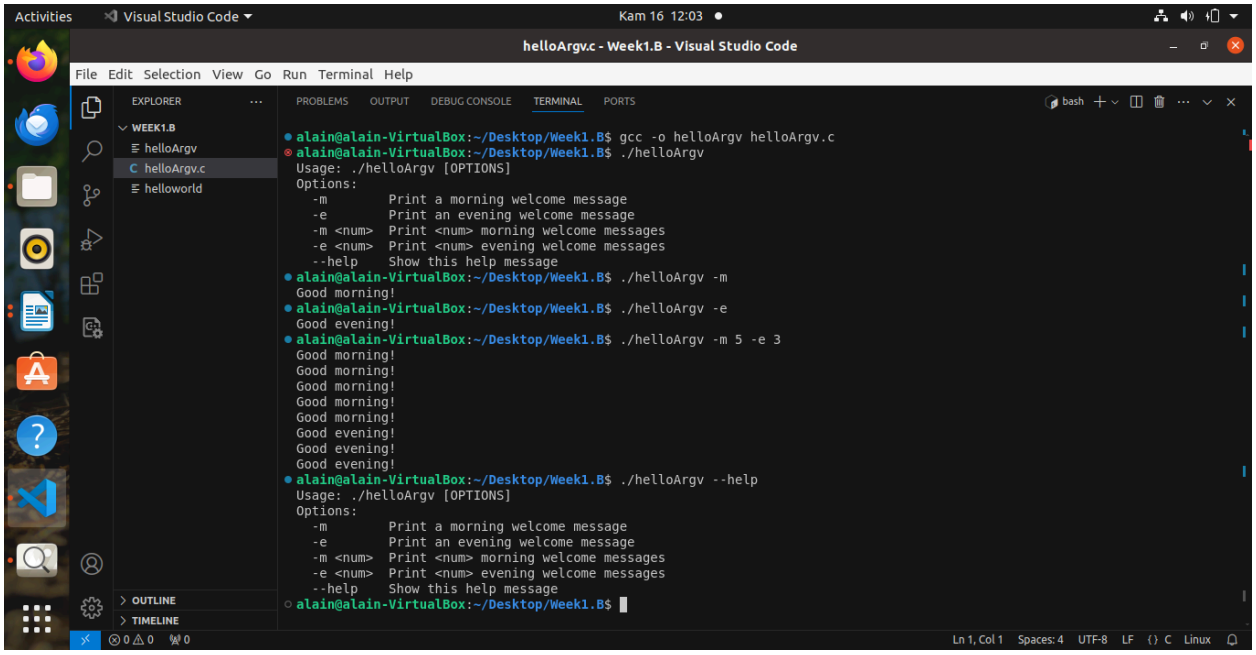


Figure3:Terminal showing different test cases and their outputs

Upon testing, the program produced the expected outputs for various command-line arguments:

1. **Basic Morning Message**

- Command: `./helloArgv -m`
- Output: `Good morning!`

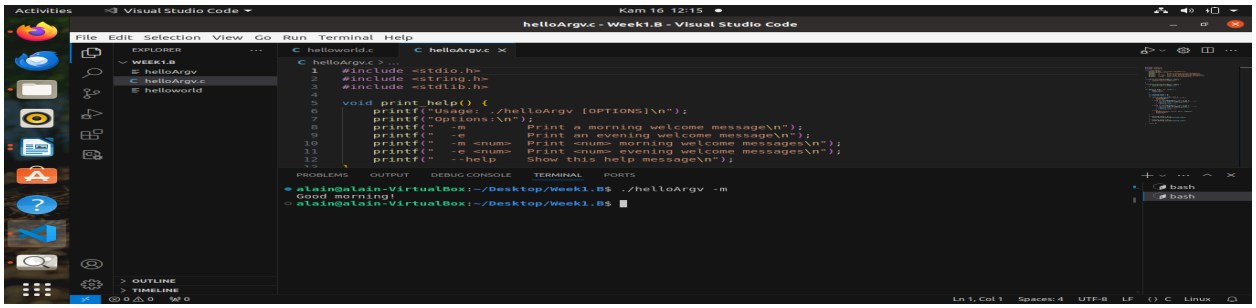


Figure 4:screenshot of terminal output for `./helloArgv -m`

2. **Basic Evening Message**

- Command: `./helloArgv -e`
- Output: `Good evening!`

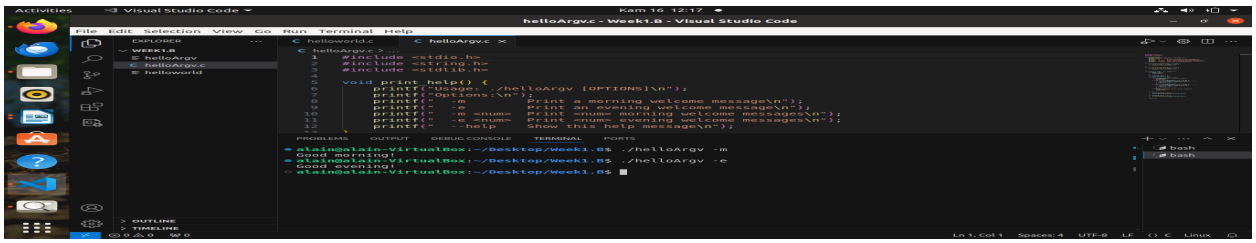


Figure5: screenshot of terminal output for `./helloArgv -e`

3. **Multiple Morning and Evening Messages**
   - Command: `./helloArgv -m 5 -e 3`
   - Output:
   ```
   Good morning!
   Good morning!
   Good morning!
   Good morning!
   Good morning!
   Good evening!
   Good evening!
   Good evening!
   ```

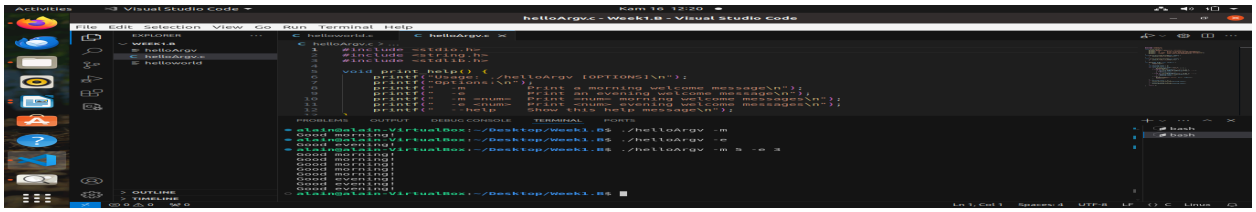


Figure6: screenshot of terminal output for `./helloArgv -m 5 -e 3`

4. **Help Message**
   - Command: `./helloArgv --help`
   - Output:
   ```
   Usage: ./helloArgv [OPTIONS]
   Options:
   -m        Print a morning welcome message
   -e        Print an evening welcome message
   -m <num>  Print <num> morning welcome messages
   -e <num>  Print <num> evening welcome messages
   --help    Show this help message
   ```

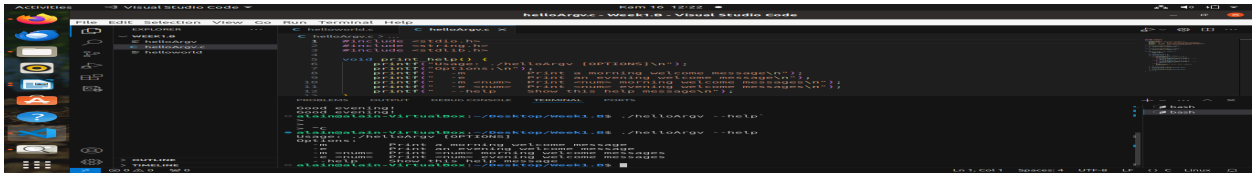


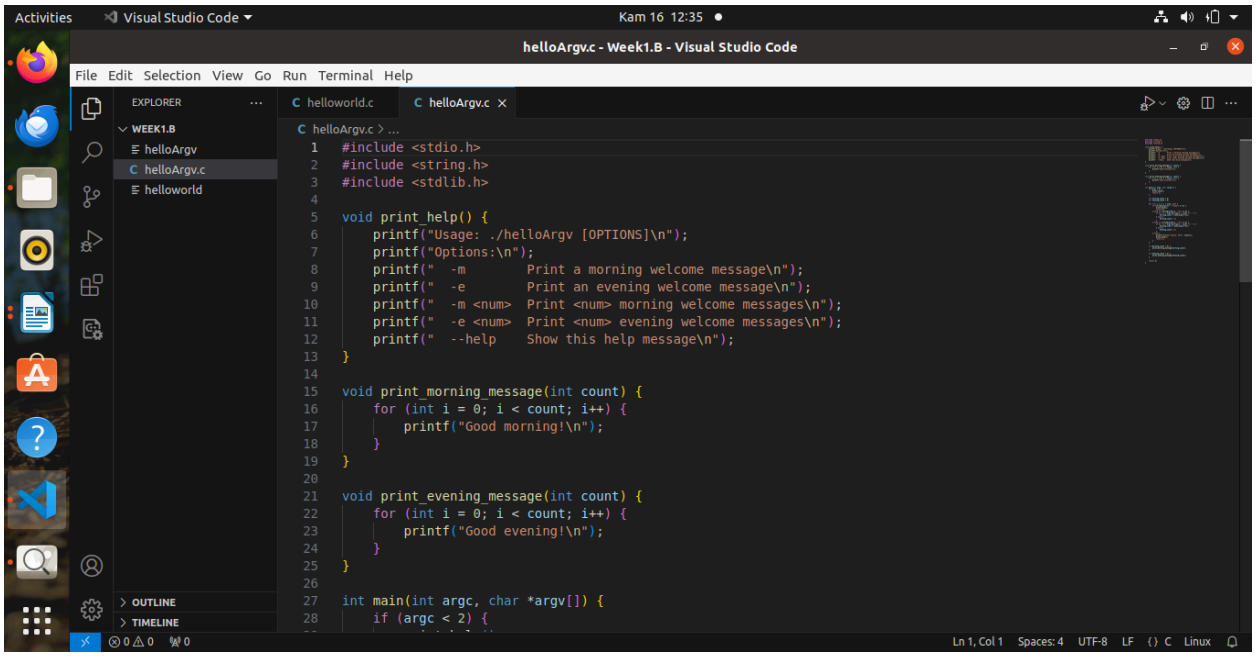
Figure 7: Insert screenshot of terminal output for `./helloArgv --help`

The program performed as expected in all test cases, fulfilling the assignment requirements.based on help expectations

Conclusion

This assignment reinforced my understanding of handling command-line arguments in C. By extending a simple program to include more complex functionality, I gained valuable experience in string comparison, conditional logic, and iterative operations. The successful completion of this task indicates a solid grasp of these concepts.

Let now delve the screenshot shows all perform code in this project



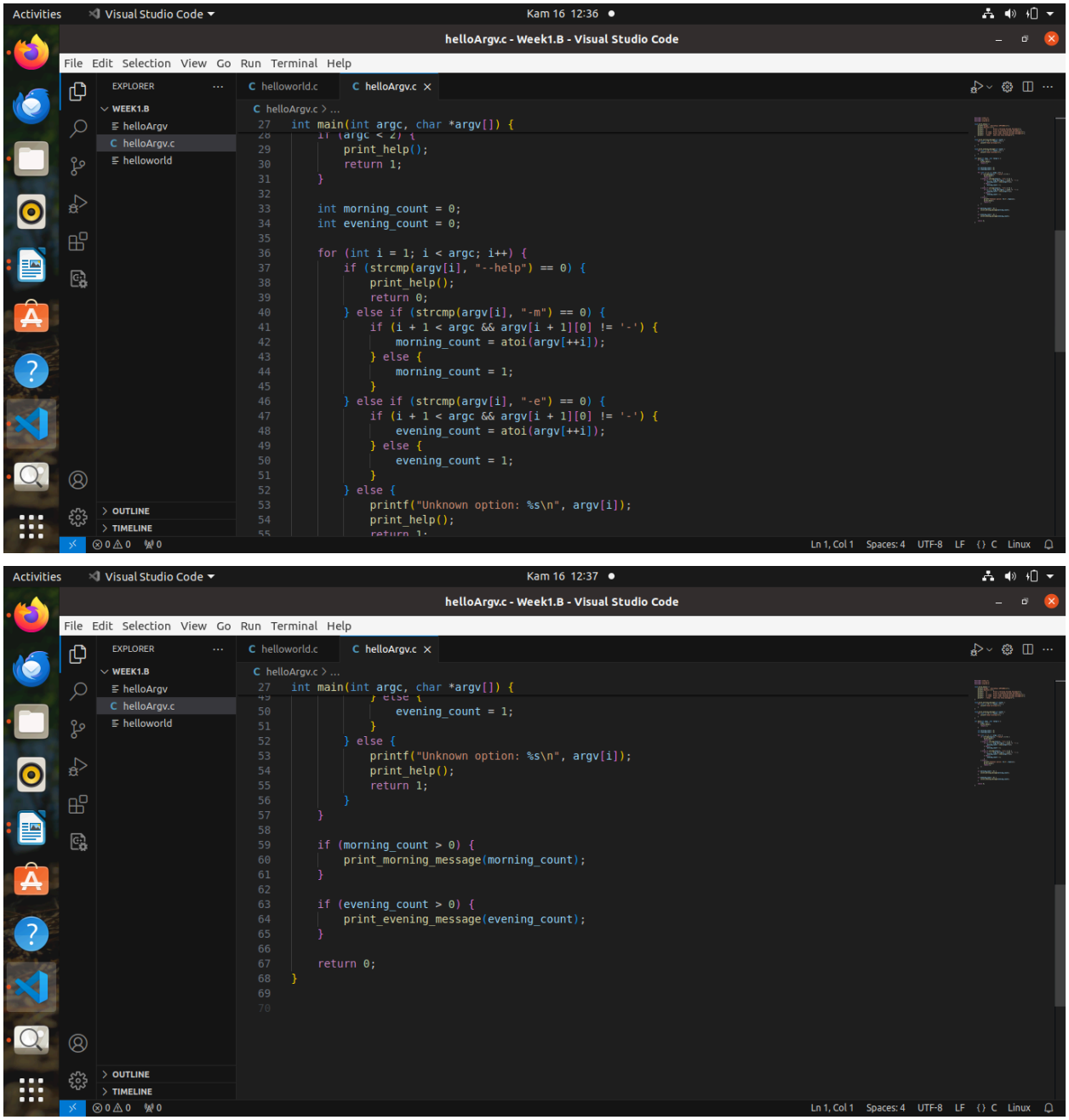

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void print_help() {
    printf("Usage: ./helloArgv [OPTIONS]\n");
    printf("Options:\n");
    printf("  -m        Print a morning welcome message\n");
    printf("  -e        Print an evening welcome message\n");
    printf("  -m <num>  Print <num> morning welcome messages\n");
    printf("  -e <num>  Print <num> evening welcome messages\n");
    printf("  --help    Show this help message\n");
}

void print_morning_message(int count) {
    for (int i = 0; i < count; i++) {
        printf("Good morning!\n");
    }
}

void print_evening_message(int count) {
    for (int i = 0; i < count; i++) {
        printf("Good evening!\n");
    }
}

int main(int argc, char *argv[]) {
    if (argc < 2) {
```




```c
int main(int argc, char *argv[]) {
    if (argc < 2) {
        print_help();
        return 1;
    }

    int morning_count = 0;
    int evening_count = 0;

    for (int i = 1; i < argc; i++) {
        if (strcmp(argv[i], "--help") == 0) {
            print_help();
            return 0;
        } else if (strcmp(argv[i], "-m") == 0) {
            if (i + 1 < argc && argv[i + 1][0] != '-') {
                morning_count = atoi(argv[++i]);
            } else {
                morning_count = 1;
            }
        } else if (strcmp(argv[i], "-e") == 0) {
            if (i + 1 < argc && argv[i + 1][0] != '-') {
                evening_count = atoi(argv[++i]);
            } else {
                evening_count = 1;
            }
        } else {
            printf("Unknown option: %s\n", argv[i]);
            print_help();
            return 1;
        }
    }

    if (morning_count > 0) {
        print_morning_message(morning_count);
    }

    if (evening_count > 0) {
        print_evening_message(evening_count);
    }

    return 0;
}
```

Figure 8: shows the all code in this program.