

# **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**

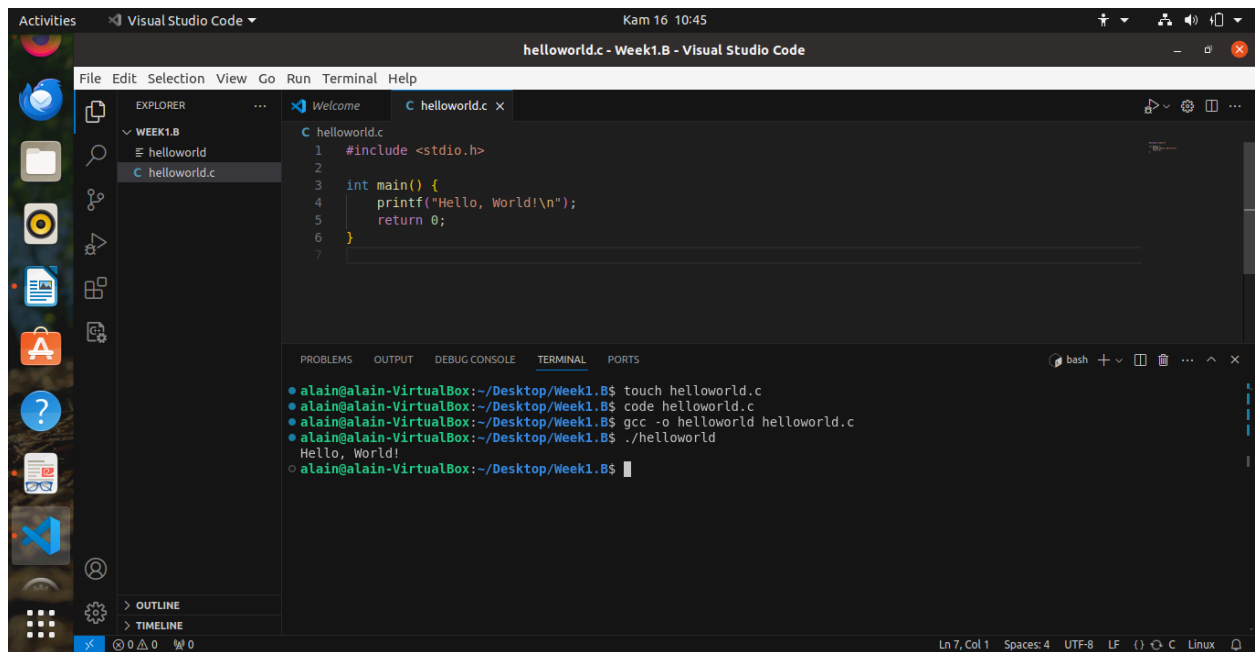
### **2. Creating and Running `helloworld.c`**

- 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:

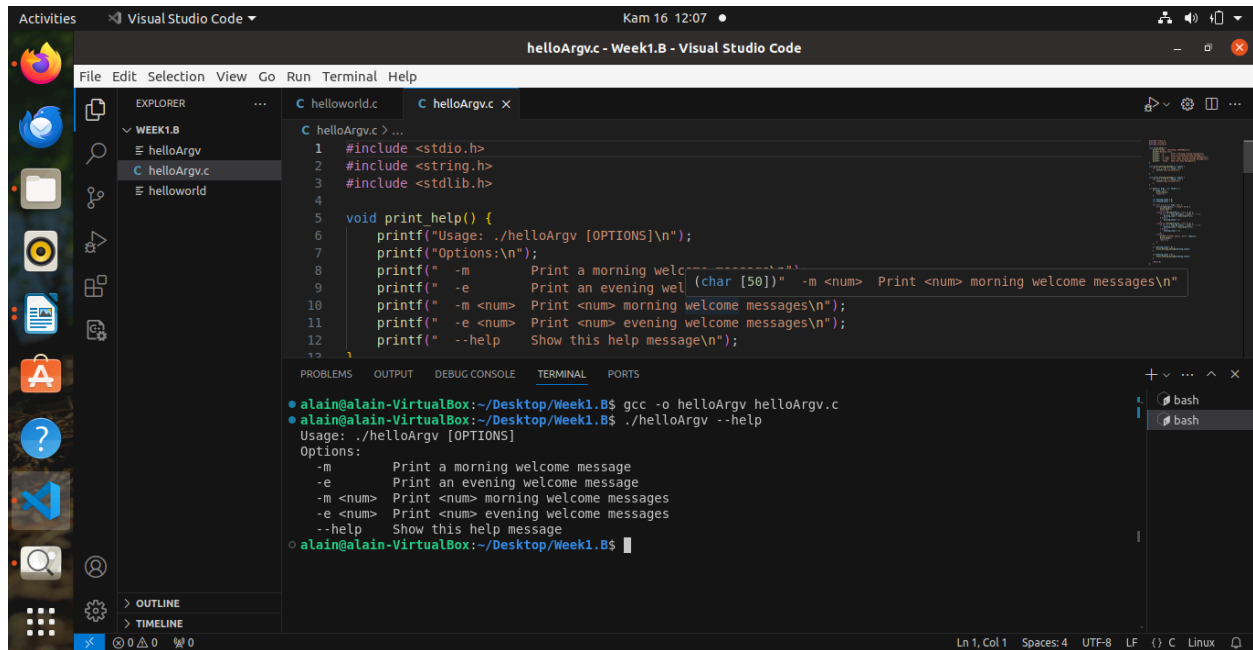
```
mv helloworld.c helloArgv.c
```

- 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.



The screenshot shows the Visual Studio Code editor with a file named 'helloArgv.c' open. The code defines a 'print\_help()' function that prints usage and options. The terminal window at the bottom shows the command './helloArgv --help' being executed, which outputs the help message defined in the code.

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 void print_help() {
6     printf("Usage: ./helloArgv [OPTIONS]\n");
7     printf("Options:\n");
8     printf("  -m      Print a morning welcome message\n");
9     printf("  -e      Print an evening welcome message\n");
10    printf("  -m <num> Print <num> morning welcome messages\n");
11    printf("  -e <num> Print <num> evening welcome messages\n");
12    printf("  --help  Show this help message\n");
13}
```

```
alain@alain-VirtualBox:~/Desktop/Week1.B$ gcc -o helloArgv helloArgv.c
alain@alain-VirtualBox:~/Desktop/Week1.B$ ./helloArgv --help
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
alain@alain-VirtualBox:~/Desktop/Week1.B$
```

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
```
```

```

alain@alain-VirtualBox:~/Desktop/Week1.B$ gcc -o helloArgv helloArgv.c
alain@alain-VirtualBox:~/Desktop/Week1.B$ ./helloArgv
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
alain@alain-VirtualBox:~/Desktop/Week1.B$ ./helloArgv -m
Good morning!
alain@alain-VirtualBox:~/Desktop/Week1.B$ ./helloArgv -e
Good evening!
alain@alain-VirtualBox:~/Desktop/Week1.B$ ./helloArgv -m 5 -e 3
Good morning!
Good morning!
Good morning!
Good morning!
Good morning!
Good evening!
Good evening!
Good evening!
alain@alain-VirtualBox:~/Desktop/Week1.B$ ./helloArgv --help
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
alain@alain-VirtualBox:~/Desktop/Week1.B$

```

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!`

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 void print_help() {
6     printf("Usage: ./helloArgv [OPTIONS]\n");
7     printf("Options:\n");
8     printf("  -m      Print a morning welcome message\n");
9     printf("  -e      Print an evening welcome message\n");
10    printf("  -m <num> Print <num> morning welcome messages\n");
11    printf("  -e <num> Print <num> evening welcome messages\n");
12    printf("  --help  Show this help message\n");
13 }
14
15 int main(int argc, char* argv[]) {
16     if (argc == 1) {
17         print_help();
18         return 1;
19     }
20     if (argc == 2) {
21         if (strcmp(argv[1], "-m") == 0) {
22             printf("Good morning!\n");
23         } else if (strcmp(argv[1], "-e") == 0) {
24             printf("Good evening!\n");
25         } else {
26             print_help();
27             return 1;
28         }
29     } else {
30         if (argc == 3) {
31             if (strcmp(argv[1], "-m") == 0) {
32                 for (int i = 0; i < atoi(argv[2]); i++) {
33                     printf("Good morning!\n");
34                 }
35             } else if (strcmp(argv[1], "-e") == 0) {
36                 for (int i = 0; i < atoi(argv[2]); i++) {
37                     printf("Good evening!\n");
38                 }
39             } else {
40                 print_help();
41                 return 1;
42             }
43         } else {
44             print_help();
45             return 1;
46         }
47     }
48     return 0;
49 }

```

```

alain@alain-VirtualBox:~/Desktop/Week1.B$ ./helloArgv -m
Good morning!
alain@alain-VirtualBox:~/Desktop/Week1.B$

```

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

2. **\*\*Basic Evening Message\*\***

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

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 void print_help() {
6     printf("Usage: ./helloArgv [OPTIONS]\n");
7     printf("Options:\n");
8     printf("  -m      Print a morning welcome message\n");
9     printf("  -e      Print an evening welcome message\n");
10    printf("  -m <num> Print <num> morning welcome messages\n");
11    printf("  -e <num> Print <num> evening welcome messages\n");
12    printf("  --help  Show this help message\n");
13 }
14
15 int main(int argc, char* argv[]) {
16     if (argc == 1) {
17         print_help();
18         return 1;
19     }
20     if (argc == 2) {
21         if (strcmp(argv[1], "-m") == 0) {
22             printf("Good morning!\n");
23         } else if (strcmp(argv[1], "-e") == 0) {
24             printf("Good evening!\n");
25         } else {
26             print_help();
27             return 1;
28         }
29     } else {
30         if (argc == 3) {
31             if (strcmp(argv[1], "-m") == 0) {
32                 for (int i = 0; i < atoi(argv[2]); i++) {
33                     printf("Good morning!\n");
34                 }
35             } else if (strcmp(argv[1], "-e") == 0) {
36                 for (int i = 0; i < atoi(argv[2]); i++) {
37                     printf("Good evening!\n");
38                 }
39             } else {
40                 print_help();
41                 return 1;
42             }
43         } else {
44             print_help();
45             return 1;
46         }
47     }
48     return 0;
49 }

```

```

alain@alain-VirtualBox:~/Desktop/Week1.B$ ./helloArgv -e
Good evening!
alain@alain-VirtualBox:~/Desktop/Week1.B$

```

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!

...

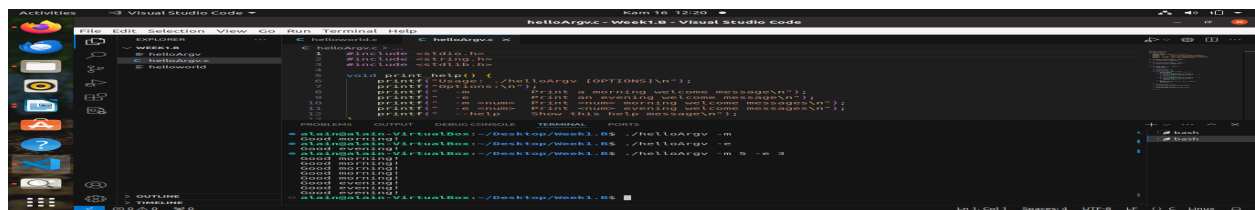


Figure6: screenshot of terminal output for `./helloArgv -m 5 -e 3`

### 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

...

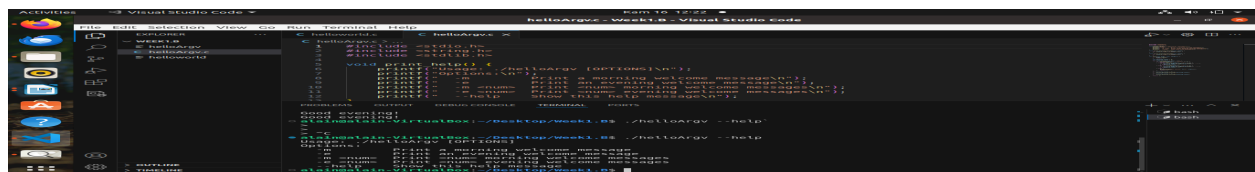


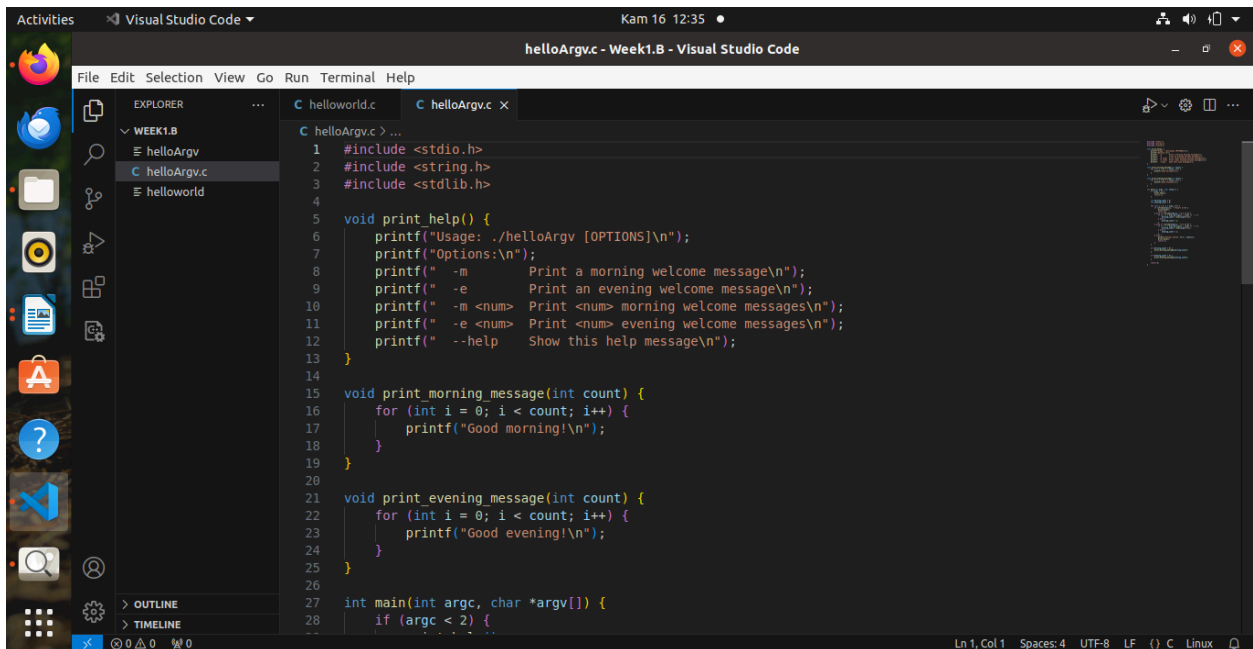
Figure 7: Insert screenshot of terminal output for `./helloArgv --help`

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



The screenshot displays the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project named 'WEEK1.B' containing files 'helloArgv', 'helloArgv.c', and 'helloworld'. The main editor window is titled 'helloArgv.c - Week1.B - Visual Studio Code' and shows the source code for 'helloArgv.c'. The code includes standard headers, defines help and message functions, and implements a main function that processes command-line arguments.

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 void print_help() {
6     printf("Usage: ./helloArgv [OPTIONS]\n");
7     printf("Options:\n");
8     printf("  -m      Print a morning welcome message\n");
9     printf("  -e      Print an evening welcome message\n");
10    printf("  -m <num> Print <num> morning welcome messages\n");
11    printf("  -e <num> Print <num> evening welcome messages\n");
12    printf("  --help  Show this help message\n");
13 }
14
15 void print_morning_message(int count) {
16     for (int i = 0; i < count; i++) {
17         printf("Good morning!\n");
18     }
19 }
20
21 void print_evening_message(int count) {
22     for (int i = 0; i < count; i++) {
23         printf("Good evening!\n");
24     }
25 }
26
27 int main(int argc, char *argv[]) {
28     if (argc < 2) {
29         print_help();
30         return 1;
31     }
32     if (strcmp(argv[1], "-m") == 0) {
33         if (argc < 3) {
34             print_help();
35             return 1;
36         }
37         print_morning_message(atoi(argv[2]));
38     } else if (strcmp(argv[1], "-e") == 0) {
39         if (argc < 3) {
40             print_help();
41             return 1;
42         }
43         print_evening_message(atoi(argv[2]));
44     } else if (strcmp(argv[1], "--help") == 0) {
45         print_help();
46     }
47     return 0;
48 }
```

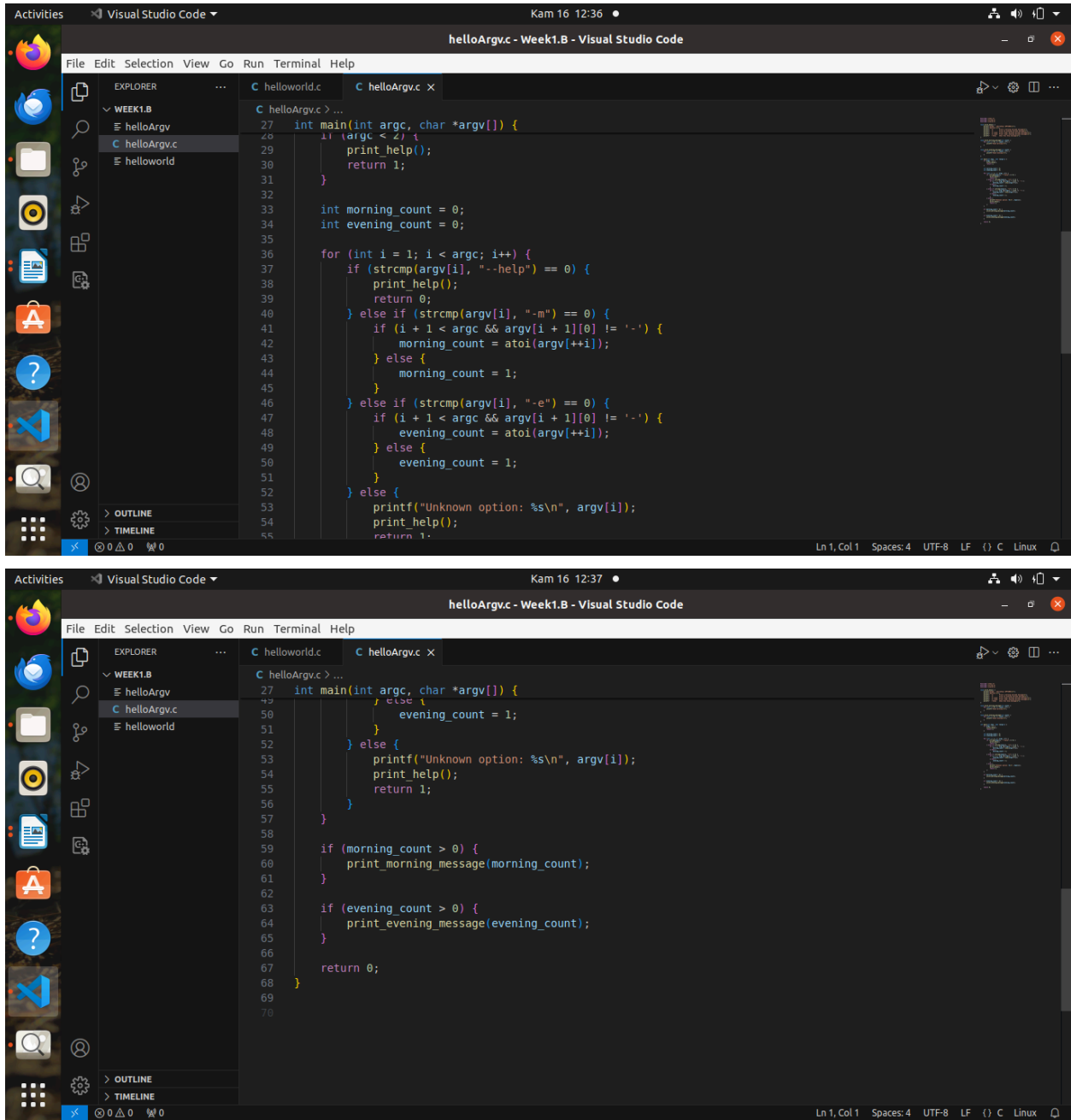


Figure 8: shows the all code in this program.