

## ASSIGNMENT 1 B REPORT

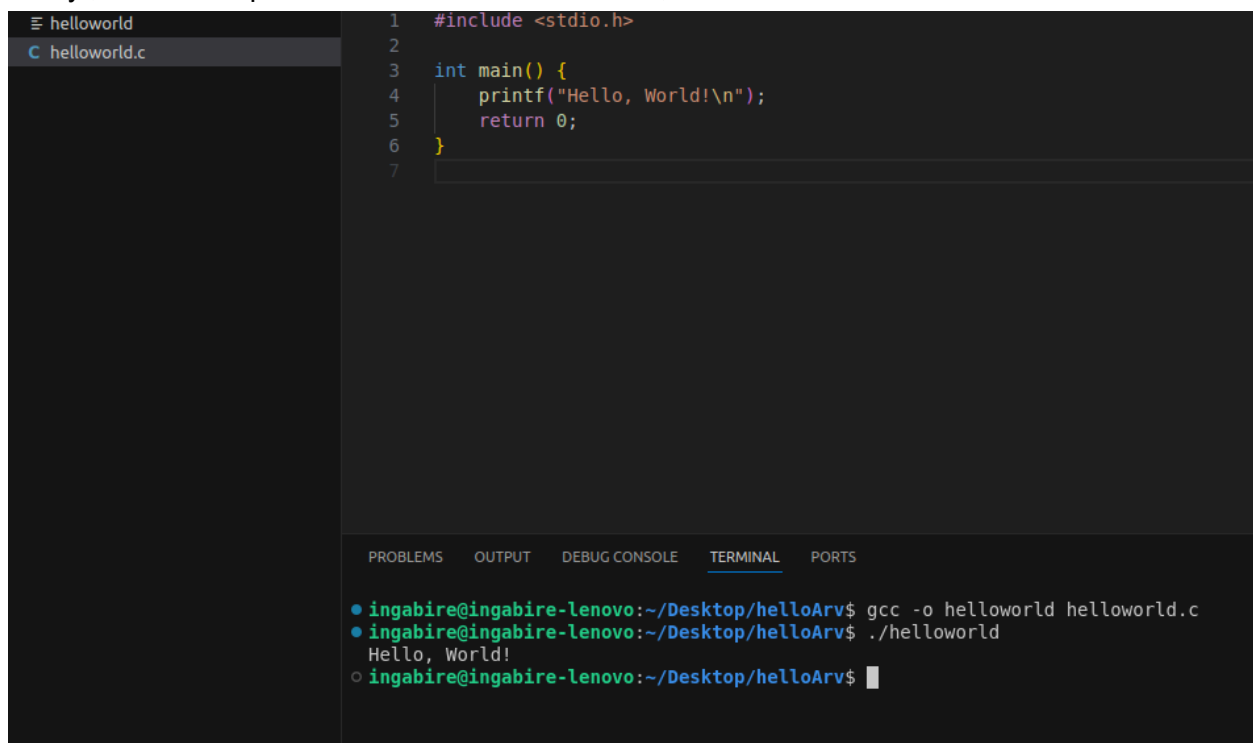
IRADUKUNDA Ingabire Delphine

### Question :helloArgv

- compile and run helloworld.c on the Host
- rename your file to helloArgv.c and extend it with the following:  
print the command line parameters (see also <http://www.thegeekstuff.com/2013/01/c-argc-argv/>)  
let the program do different things dependent of the parameters, e.g.
  1. -m prints a morning-welcome-message
  2. -e prints an evening-welcome-message
  3. -m 5 -e 3 prints five morning and three evening messages;
  4. and of course you provide a --help.

### Answer:

Firstly , I have compile and run helloworld.c codes , here is the screenshot above



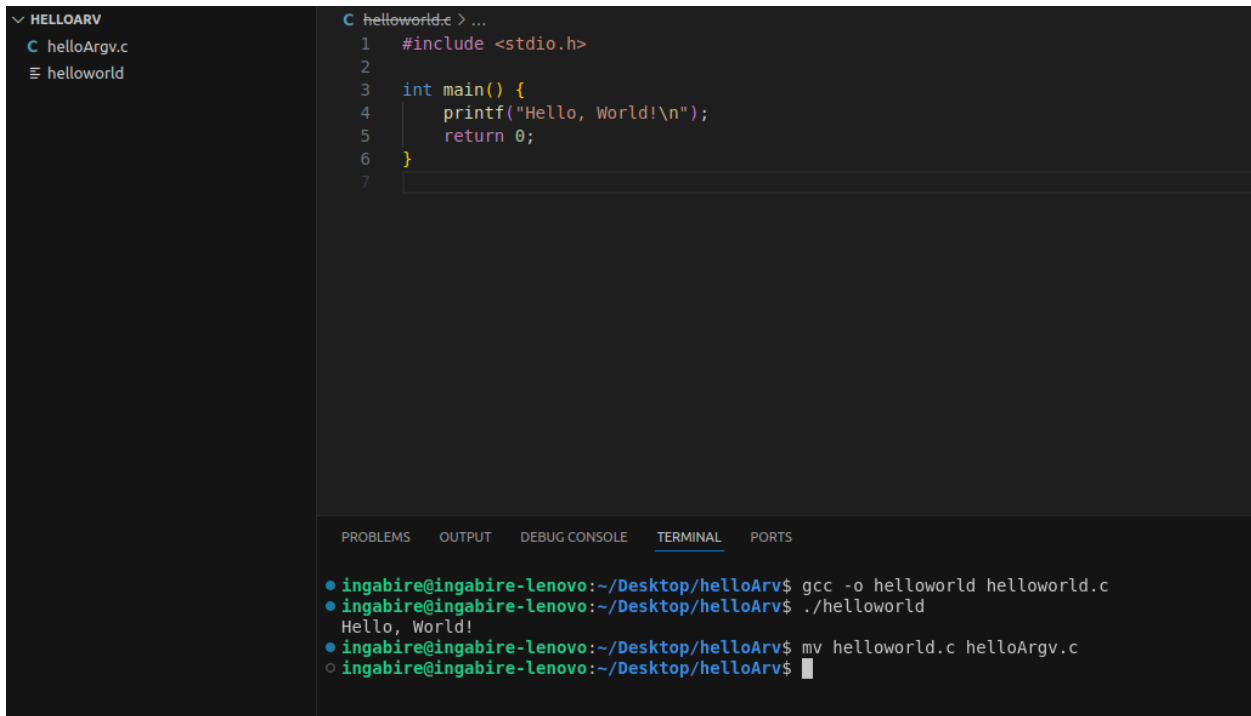
The screenshot shows a code editor with a file named `helloworld.c` containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello, World!\n");
5     return 0;
6 }
7
```

Below the code editor is a terminal window with the following output:

```
ingabire@ingabire-lenovo:~/Desktop/helloArv$ gcc -o helloworld helloworld.c
ingabire@ingabire-lenovo:~/Desktop/helloArv$ ./helloworld
Hello, World!
ingabire@ingabire-lenovo:~/Desktop/helloArv$
```

And as i have told in the question i have renamed the file from helloworld.c to helloArgv.c



```
HELLOARV
C helloworld.c
helloworld

C helloworld.c > ...
1  #include <stdio.h>
2
3  int main() {
4      printf("Hello, World!\n");
5      return 0;
6  }
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

• ingabire@ingabire-lenovo:~/Desktop/helloArv$ gcc -o helloworld helloworld.c
• ingabire@ingabire-lenovo:~/Desktop/helloArv$ ./helloworld
Hello, World!
• ingabire@ingabire-lenovo:~/Desktop/helloArv$ mv helloworld.c helloArgv.c
○ ingabire@ingabire-lenovo:~/Desktop/helloArv$
```

As you see in the picture above i have renamed helloworld.c to helloArgv.c and i am going to put the code of helloArgv.c compile it and also run it.

Here is my answer of helloArgv.c :

**CODES:**



```
HELLOARV
helloArgv
C helloArgv.c

C helloArgv.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  // Function to print the usage/help message
6  void print_usage(const char *progname) {
7      printf("Usage: %s [OPTIONS] [MESSAGE_COUNT]\n", progname);
8      printf("Options:\n");
9      printf("  -m, --morning   Print a morning welcome message.\n");
10     printf("  -e, --evening   Print an evening welcome message.\n");
11     printf("  -h, --help      Display this help message.\n");
12     printf("[MESSAGE_COUNT] (optional): Number of messages to print (defaults to 1).\n");
13 }
14
15 // Function to print morning messages
16 void print_morning_message(int count) {
17     for (int i = 0; i < count; ++i) {
18         printf("Good morning!\n");
19     }
20 }
21
22 // Function to print evening messages
23 void print_evening_message(int count) {
24     for (int i = 0; i < count; ++i) {
25         printf("Good evening!\n");
26     }
27 }
28
29 int main(int argc, char *argv[]) {
30     int want_morning = 0, want_evening = 0;
31     int morning_count = 1, evening_count = 1;
32     int message_count = 1;
33
34     // Process command-line arguments
35     for (int i = 1; i < argc; ++i) {
36         if (strcmp(argv[i], "-m") == 0 || strcmp(argv[i], "--morning") == 0) {
37             want_morning = 1;
38         } else if (strcmp(argv[i], "-e") == 0 || strcmp(argv[i], "--evening") == 0) {
39             want_evening = 1;
40         } else if (strcmp(argv[i], "-h") == 0 || strcmp(argv[i], "--help") == 0) {
41             print_usage(argv[0]);
42             return 0;
43         } else if (i == argc - 1) {
44             message_count = atoi(argv[i]);
45         }
46     }
47
48     if (want_morning) {
49         print_morning_message(message_count);
50     }
51     if (want_evening) {
52         print_evening_message(message_count);
53     }
54
55     return 0;
56 }
```

```
29 int main(int argc, char *argv[]) {
34     // Process command-line arguments
35     for (int i = 1; i < argc; ++i) {
36         if (strcmp(argv[i], "-m") == 0 || strcmp(argv[i], "--morning") == 0) {
37             want_morning = 1;
38             if (i + 1 < argc && sscanf(argv[i + 1], "%d", &message_count) == 1) {
39                 morning_count = message_count;
40                 i++; // Skip the count argument
41             }
42         } else if (strcmp(argv[i], "-e") == 0 || strcmp(argv[i], "--evening") == 0) {
43             want_evening = 1;
44             if (i + 1 < argc && sscanf(argv[i + 1], "%d", &message_count) == 1) {
45                 evening_count = message_count;
46                 i++; // Skip the count argument
47             }
48         } else if (strcmp(argv[i], "-h") == 0 || strcmp(argv[i], "--help") == 0) {
49             print_usage(argv[0]);
50             return 0;
51         } else {
52             fprintf(stderr, "Error: Unknown option '%s'.\n", argv[i]);
53             print_usage(argv[0]);
54             return 1;
55         }
56     }
57
58     // Validate and handle errors
59     if (!want_morning && !want_evening) {
60         fprintf(stderr, "Error: You must specify either morning or evening.\n");
61         return 1;
62     }
63
64     // Print messages based on choices and count
65     if (want_morning) {
66         print_morning_message(morning_count);
67     }
68     if (want_evening) {
69         print_evening_message(evening_count);
70     }
71
72     return 0;
73 }
74
```

Here is the explanation of the above codes:

I have begin the program begins by including three standard libraries: **stdio.h** for input and output operations, **stdlib.h** for memory allocation and process control, and **string.h** for string handling functions.

A **print\_usage** function is defined to display usage information to the user, detailing the program's expected usage, available options, and their descriptions. This function takes the program name as an argument and prints the expected usage of the program. Two additional functions, **print\_morning\_message** and **print\_evening\_message**, are defined to print "Good morning!" and "Good evening!" messages respectively, a specified number of times based on the provided count.

The **main** function serves as the entry point of the program. It initializes several variables: **want\_morning** and **want\_evening** as flags to indicate whether the morning or evening message should be printed, and **morning\_count** and **evening\_count** as counters for how many times to print each message. A temporary variable, **message\_count**, is used to parse the count of messages from the command line. The program then processes the command-line arguments starting from index 1 (since index 0 is the program name). It checks if the argument is **-m** or **--morning**, setting the **want\_morning** flag to 1 and updating **morning\_count** if the next argument is a valid integer. Similarly, it checks for **-e** or **--evening**, setting the **want\_evening** flag to 1 and updating **evening\_count** if the next argument is a valid integer. If the argument is **-h** or **--help**, it calls the **print\_usage** function and exits. For any unknown option, it prints an error message and the usage information.

The program validates the inputs by checking if neither **want\_morning** nor **want\_evening** is set, in which case it prints an error message and exits. Finally, based on the flags set, the program calls the appropriate message printing function (**print\_morning\_message** or **print\_evening\_message**) with the specified count.

## OUTPUT:

```
✓ HELLOARV
  helloArgv
  C helloArgv.c

● ingabire@ingabire-lenovo:~/Desktop/helloArv$ gcc -o helloArgv helloArgv.c
● ingabire@ingabire-lenovo:~/Desktop/helloArv$ ./helloArgv -m 5
Good morning!
Good morning!
Good morning!
Good morning!
Good morning!
● ingabire@ingabire-lenovo:~/Desktop/helloArv$ ./helloArgv -e 3
Good evening!
Good evening!
Good evening!
● ingabire@ingabire-lenovo:~/Desktop/helloArv$ ./helloArgv -e
Good evening!
● ingabire@ingabire-lenovo:~/Desktop/helloArv$ ./helloArgv -m
Good morning!
● ingabire@ingabire-lenovo:~/Desktop/helloArv$ ./helloArgv -m 5 -e 3
Good morning!
Good morning!
Good morning!
Good morning!
Good morning!
Good evening!
Good evening!
Good evening!
○ ingabire@ingabire-lenovo:~/Desktop/helloArv$
```