PARADOX TECHNOLOGY HUB

ASSIGNMENT 2 FOR WEEK1

WEEK1 REPORT

Robotic Team

Question1

- 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.

    `-m` prints a *morning*-welcome-message

    `-e` prints an *evening*-welcome-message

    `-m 5 -e 3` prints five morning and three evening messages;

    and of course you provide a `-help`

Answer;

Step1: Creation and compilation of 'helloworld.c'

1. Create a file 'helloworld.c'

```c
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

2. Compilation of 'helloworld.c'

```
gcc helloworld.c -o helloworld
```

3. Running the compiled 'helloworld.c'

```
./helloworld
```

4. Expected output

```
Hello, World!
```

Step2: Rename and extend the program to ‘helloArgv.c’

1. Renaming the file ‘helloworld.c’ to ‘helloArgv.c’

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

2. Extend ‘helloArgv.c’ to handle command-line arguments

```
#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     Display this help message\n");
}

int main(int argc, char *argv[]) {
    if (argc == 1) {
        printf("Hello, World!\n");
        return 0;
    }
```

```
    int i, morning_count = 0, evening_count = 0;

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

```
    }

    for (i = 0; i < morning_count; i++) {
        printf("Good morning!\n");
    }

    for (i = 0; i < evening_count; i++) {
        printf("Good evening!\n");
    }

    return 0;
}
```

3. Compilation of ‘helloArgv.c’

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

4. Run the extended program with various arguments

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

5. Expected Outputs

- **`./helloArgv --help`**:

mathematica

```
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      Display this help message
```

- **`./helloArgv -m`**:

Copy code

```
Good morning!
```

- **`./helloArgv -e`**:

Copy code

```
Good evening!
```

- `./helloArgv -m 5 -e 3`:

Copy code

```
Good morning!
Good morning!
Good morning!
Good morning!
Good morning!
Good evening!
Good evening!
Good evening!
```

Explanation

Header Inclusions

c Copy code

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

These lines include necessary libraries:

- `stdio.h` for input and output functions (like `printf`).
- `string.h` for string handling functions (like `strcmp`).
- `stdlib.h` for utility functions (like `atoi`).

Function to Print Help

c Copy code

```c
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      Display this help message\n");
}
```

This function prints out instructions on how to use the program, including the available command-line options.

Main Function

c Copy code

```c
int main(int argc, char *argv[]) {
```

The `main` function starts here. It takes two arguments:

- `argc`: The count of command-line arguments.
- `argv`: An array of strings (character pointers) representing the arguments.

No Arguments Case

c Copy code

```c
    if (argc == 1) {
        printf("Hello, World!\n");
        return 0;
    }
```

If no command-line arguments are provided (`argc == 1`), the program prints "Hello, World!" and exits

Variable Initialization

c Copy code

```c
int i, morning_count = 0, evening_count = 0;
```

These variables are initialized:

- `i` for loop iteration.
- `morning_count` and `evening_count` to store the number of times morning and evening messages should be printed.

Argument Parsing Loop

c Copy code

```c
for (i = 1; i < argc; i++) {
```

This loop iterates over each command-line argument, starting from index 1 (since index 0 is the program's name).

Handling --help Option

c Copy code

```c
if (strcmp(argv[i], "--help") == 0) {
    print_help();
    return 0;
```

If `--help` is found, the `print_help` function is called, and the program exits.

Handling -m Option

c Copy code

```c
} else if (strcmp(argv[i], "-m") == 0) {
    if (i + 1 < argc && isdigit(argv[i + 1][0])) {
        morning_count = atoi(argv[i + 1]);
        i++;
    } else {
        morning_count = 1;
    }
```

If `-m` is found, the program checks if the next argument is a digit. If it is, `morning_count` is set to the provided number, and `i` is incremented to skip the number. If no number is provided, `morning_count` is set to 1.

Handling -e Option

c Copy code

```c
} else if (strcmp(argv[i], "-e") == 0) {
    if (i + 1 < argc && isdigit(argv[i + 1][0])) {
        evening_count = atoi(argv[i + 1]);
        i++;
    } else {
        evening_count = 1;
    }
```

If `-e` is found, the program checks if the next argument is a digit. If it is, `evening_count` is set to the provided number, and `i` is incremented to skip the number. If no number is provided, `evening_count` is set to 1.

Handling Unknown Option

```c
        } else {
            printf("Unknown option: %s\n", argv[i]);
            print_help();
            return 1;
        }
    }
```

If an unknown option is found, an error message is printed, the help message is displayed, and the program exits with an error code.

Printing Messages

```c
    for (i = 0; i < morning_count; i++) {
        printf("Good morning!\n");
    }

    for (i = 0; i < evening_count; i++) {
        printf("Good evening!\n");
    }
```

After parsing the arguments, the program prints the appropriate number of morning and evening messages based on the values of `morning_count` and `evening_count`.

Returning from Main

```c
    return 0;
}
```

Finally, the program exits successfully.