Week 06, Lab 06 **Weight: 1%** **Due: End of your stream's week 6 lab session (via `sync`)**

Pre-lab Preparation:

- **Week 1, 2, 3, 4, 5 Lectures, Week 1, 2, 3, 4, 5 Labs**
- **Lecture: Week 06 (015, 016)**

Lab Activities:

Remember to **`sync`** to obtain the lab starting code.

Exercise 1: Learning to Unzip

Navigate to the directory: **`~/p1.2015s1/lab06/working_copy/`**

Inside this directory you will find a zip file called **`ex01.zip`**. A zip is an archive file that can compress multiple files into a single zip file. Use **`ls`** to check the file is present in the directory.

You must extract the zip to decompress the files within it. To achieve this in Linux, type:

```
unzip -P <password> lab01.zip
```

As practice for the test you will be given the password during the lab session by the lab TA.

Now, use **`ls`** to view the contents of the **`working_copy`** directory. Notice there are files now inside the directory; this is what was unzipped.

Exercise 2: How to prepare for the Mid-Semester Test

Navigate to the directory: **`~/p1.2015s1/lab06/working_copy/ex02/`**

The lab TA will also provide you with a printed hand out of “Programming 1 Test and Exam and Preparation Guide”. Review this document.

You will want to prepare your own unique materials for the open book test. Ensure you set aside time during your self-directed study this week to prepare your notes for the test, and complete any further study you require in preparation.

Complete the electronic Week 1 to Week 5 Knowledge Checklist, open **`checklist.txt`** in the **`editor`**. Once you have completed the checklist. Run a **`sync`**.

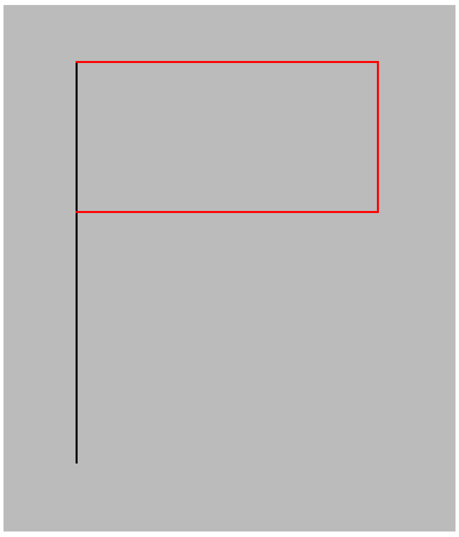
Once complete, have a TA review your completed exercises 1 and 2 for this lab session. See the end of this document for the review questions.

Exercise 3: Sample Question

Navigate to the directory: **`~/p1.2015s1/lab06/working_copy/ex03/`**

This is an easy Turtle World drawing exercise.

Using the Turtle World, write a **`void`** function called **`draw_flag()`** that draws the following:



Call **`draw_flag`** from your **`main`** function, also ensure you create a turtle world and shutdown the P1 world!

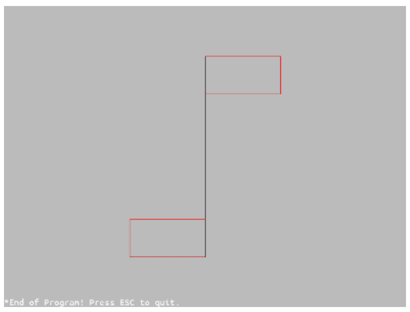
Ensure you complete the file header comment accurately.

Exercise 4: Sample Question

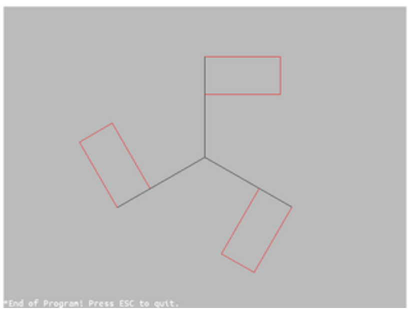
Navigate to the directory: **`~/p1.2015s1/lab06/working_copy/ex04/`**

This is a slightly harder Turtle World exercise, it uses repetition as well as functions with parameters.

Using the Turtle World, write a **`void`** function called **`draw_n_flags(int n)`** that draws *n* number of flags (as in exercise 4) in the following style:

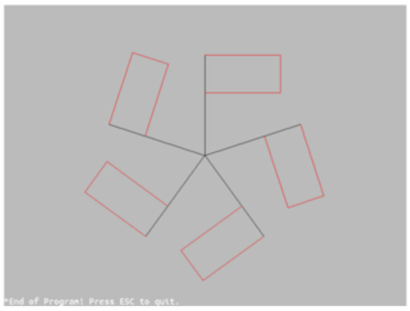


```
*End of Program! Press ESC to quit.
```

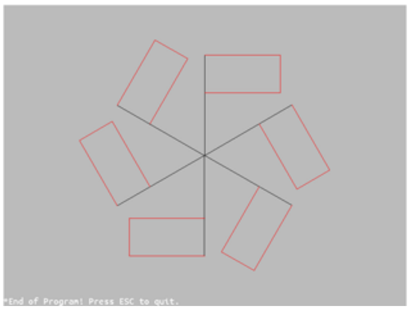


```
*End of Program! Press ESC to quit.
```

Left Above: n is 2, Right Above: n is 3.

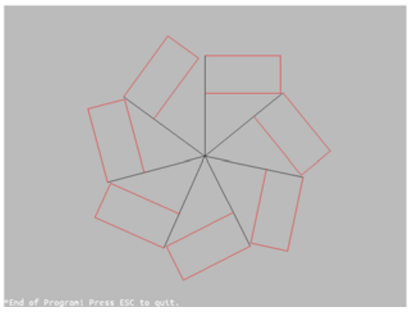


```
*End of Program! Press ESC to quit.
```

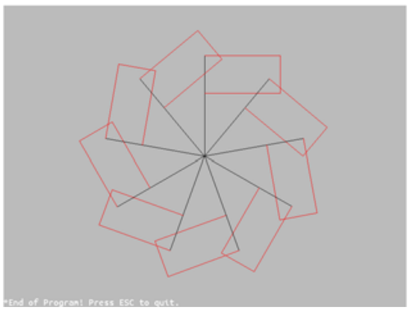


```
*End of Program! Press ESC to quit.
```

Left Above: n is 5, Right Above: n is 6.

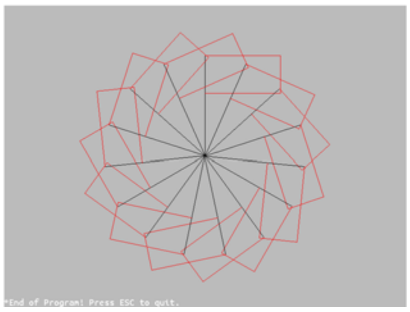


```
*End of Program! Press ESC to quit.
```

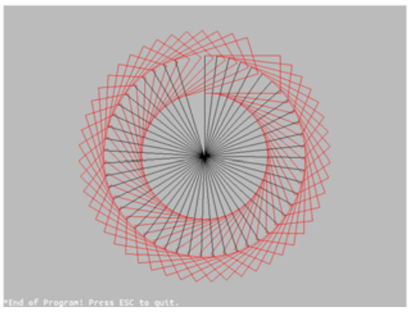


```
*End of Program! Press ESC to quit.
```

Left Above: n is 7, Right Above: n is 9.



```
*End of Program! Press ESC to quit.
```



```
*End of Program! Press ESC to quit.
```

Left Above: n is 15, Right Above: n is 50.

Once you have completed your **`draw_n_flags(int n)`** function, simply call it once from the **`main`** function with a literal **`5`** passed in as the argument.

Ensure you complete the file header comment accurately.

Exercise 5: Sample Question

This is an example of an easier kind of question that might appear in the test.

In this question the robot starts in position (1, 1) facing east. A stack of beepers will be placed either to the north or to the east. If the beepers are to the east then the robot should pick them all up and lay them out in a line west to east. If they are to the north then the robot should lay them out in a line south to north.

Implement this algorithm by writing the following functions:

```
void find_beepers()
```

The robot should move to the east and check if the beepers are present. IF they are NOT it should turn around, move east, and then move north. The beepers are guaranteed to be either north or east so no further check is required.

```
void pick_up_stack()
```

WHILE there is an item on the floor at the robot the robot should pick up an item.

```
void lay_out_stack()
```

WHILE the robot is holding an item the robot should drop an item and move forwards one square. It is guaranteed that there will not be so many items that the robot would crash into a wall.

```
void spread_beepers()
```

By calling previously defined functions, implement the algorithm.

Make sure all tests pass when you run the following command.

```
make test
```

You can also use make test to examine possible starting scenarios by defining a dummy function for **`spread_beepers()`** first.

Exercise 6: Sample Questions

This is an example of a harder kind of question that might appear in the test.

In this question the robot starts inside a room. There is a door located at random in one of the four walls, with a beeper outside it. The robot must find the door and walk to the beeper.

Implement an algorithm in the function…

```
void escape_room()
```

…to solve this approach, defining appropriate additional functions and comments to make clear the steps involved.

Make sure all tests pass when you run the following command.

```
make test
```

You can also use make test to examine possible starting scenarios by defining a dummy function for **`spread_beepers()`** first.

Exercise 7: Sample Questions

Navigate to the directory: **`~/p1.2015s1/lab06/working_copy/ex07/`**

This is a slightly tricky question on output, it requires you know how to print to the console, and what escape character codes are.

Using **`stdio.h`** and **`printf`**, write a console program which outputs the following:

```
+-----------+
|           |
|  /\   /\  |
|  \/   \/  |
|           |
|  [-=-=-]  |
|           |
+-----------+
```

Beware of the escape sequence character codes! Ensure you output exactly matches the desired output above. You should write a void function called **`print_robot_face()`** which when called from the main function prints the robot face!

Exercise 8: Sample Questions

Navigate to the directory: **`~/p1.2015s1/lab06/working_copy/ex08/`**

This is an easy console input and output question which covers variable usage and printing and reading data from the console.

Write a program which asks the user for two whole numbers, and then outputs the addition, subtraction, multiplication, integer division and modulus of the two numbers.

Your program must function as followings:

```
Hello User!
Please enter the first number: 50
Please enter the second number: 8
Calculating...
50 plus 8 is 58
50 minus 8 is 42
50 times 8 is 400
50 divided by 8 is 6
50 modulus 8 is 2
Thank you! Goodbye!
```

Ensure you output style exactly matches the desired output above, and that the program works for any whole numbers input, not just the sample 50 and 8 as shown above!

Exercise 9: Sample Questions

Navigate to the directory: `~/p1.2015s1/lab06/working_copy/ex09/`

This is a slightly trickier question which requires you to use variables and make selections based upon user input. You may need to use logical operators, or nested block statements to achieve the desired result.

Write a console program which queries the user as to whether it is a weekday, and if it is sunny.

Based upon the user's input, the program must output the following reminders:

- On sunny weekdays the program must remind the user to wear sunblock.
- On sunny weekends the program must remind the user to go to the beach.
- On non-sunny weekdays the program must remind the user to take an umbrella.
- On non-sunny weekends the program must remind the user to stay home and play board games.

Your program must function as followings:

```
Welcome to Reminder-Bot-2K15!

Is it sunny outside (y/n)? n
Is it a weekday (y/n)? n

Reminder-Bot-2K15 says: "STAY-HOME-AND-PLAY-BOARD-GAMES!"

Thank you for using Reminder-Bot-2K15!
```

Ensure your program functions correctly with all possible `y` or `n` input combinations. Also complete the file header comment accurately.

Week 06, Lab 06 Submission:

Run the **sync** command to submit your completed lab work.

Shutdown your Raspberry PI by pressing **ALT-CTRL-DEL**. Power-down and pack up your Raspberry Pi kit.

Marking Criteria:

Have you completed each of the following? Have you submitted your code from lab?

Marking Criteria:	**Week 06 Lab 06 Weight 1%**	**Yes**	**No**
Ex 1:	Password protected zip successfully extracted?		
Ex 2:	Knowledge Checklist completed and sync'd?		
Ex 3:	draw_flag called from main and appropriately draws the required output?		
Ex 4:	draw_n_flags(5) called from main and appropriately draws the required output?		
Ex 5:	All tests pass?		
	Algorithm structure appropriately?		
	Good indentation practices used?		
Ex 6:	All tests pass?		
	Algorithm structure appropriately?		
	Good indentation practices used?		
Ex 7:	print_robot_face() called from main and accurately draws the required output?		
Ex 8:	Program correctly calculates outputs the addition, subtraction, multiplication, division and modulus for any given input?		
Ex 9:	Program functions correctly with all possible **y** or **n** input combinations.		
	File header comment completed correctly?		

Next activity: Mid-Semester Test Preparation and Final Week 6 Lecture