LED Worksheet #1

1. Type in this missing parts
Beware of the upper and lower case!

```cpp
#include <EngduinoLEDs.h>

void setup() {
  // put your setup code here, to run once:
  EngduinoLEDs.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  EngduinoLEDs.setAll(RED);
}
```

2. Press this button to “upload” to Engduino

3. Other colours to try (all upper case):

   - RED
   - GREEN
   - BLUE
   - YELLOW
   - MAGENTA
   - CYAN
   - WHITE
   - OFF

These are all the colours we defined. But we can mix more!
LED Worksheet #2

1. Use RGB (*Red*, *Green* and *Blue*) to make a new colour. Mix your own colour.

   ```cpp
   #include <EngduinoLEDs.h>

   void setup() {
     // put your setup code here, to run once:
     EngduinoLEDs.begin();
   }

   void loop() {
     // put your main code here, to run repeatedly:
     EngduinoLEDs.setAll(100, 50, 2);
   }
   
   The number goes from 0 to 255. 0 means the colour is off. 255 is the maximum brightness. Change these numbers to mix your own colour.

2. Press this button to “upload” to Engduino.

   Can you mix a pink colour?
#include <EngduinoLEDs.h>

void setup() {
  // put your setup code here, to run once:
  EngduinoLEDs.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  EngduinoLEDs.setLED(0, RED);
  EngduinoLEDs.setLED(1, GREEN);
  EngduinoLEDs.setLED(2, BLUE);
  EngduinoLEDs.setLED(3, YELLOW);
  EngduinoLEDs.setLED(4, GREEN);
  EngduinoLEDs.setLED(5, MAGENTA);
}

1. Now we learn to make a sequence of lights

2. Press this button to “upload” to Engduino

We have 16 LEDs. Number 0 to 15. This is because in programming, we usually starts at 0.
1. Let’s do some blinking

```c
#include <EngduinoLEDs.h>

void setup() {
  // put your setup code here, to run once:
  EngduinoLEDs.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  EngduinoLEDs.setAll(YELLOW);
  delay(1000);
  EngduinoLEDs.setAll(OFF);
  delay(1000);
  EngduinoLEDs.setAll(BLUE);
  delay(1000);
  EngduinoLEDs.setAll(OFF);
  delay(1000);
}
```

2. Press this button to “upload” to Engduino

delay(1000) means delay 1 second (1000 milliseconds)

Why don’t you try making it blink faster or slower?
LED Worksheet #5

Here are some programming tricks:-

#include <EngduinoLEDs.h>

void setup() {
    // put your setup code here, to run once:
    EngduinoLEDs.begin();
}

int DD = 500;

void loop() {
    // put your main code here, to run repeatedly:
    for (int i = 0; i < 16; i++) {
        EngduinoLEDs.setLED(i, BLUE);
        delay(DD);
        EngduinoLEDs.setAll(OFF);
    }
}

**Variable**

int = integer. We make a “variable” which has a type integer and name it DD. A variable stores a constant value or text. We can change the value of this variable in the program. Also, if we use variable in the program. If we decide to change the value later, we only need to change it in one place instead of many places.

**For-loop**

In a for-loop, we define the start condition, the stop condition and step condition each time we go through the for loop. In this example, I would start from 0, each time we go through the for loop, it will add one to i until i=15. i is also a variable! You can see that the value i changes in the program!

**Scope**

{} define the scope of our setup(), loop() and for-loop. Beware, the number of { open brackets must match the number of } close brackets!
#include <EngduinoLEDs.h>

void setup() {
  // put your setup code here, to run once:
  EngduinoLEDs.begin();
}

int DD = 500;

void loop() {
  // put your main code here, to run repeatedly:

  for (int i = 0; i < 16; i++) {
    EngduinoLEDs.setLED(i, BLUE);
    delay(DD);
    EngduinoLEDs.setAll(OFF);
  }
  //decrease the delay time
  DD -= 50;
  if (DD <= 0) {
    DD = 500;
  }
}

**If-statement**
The “If-statement” in our example resets the value of DD if it is smaller than or equal to 0. So our lights continue to go round faster and faster repeatedly.

**Now it is your turn to make a light display.**