CSE101 F24 – Assignment 1 - Çağatay Kağan Atalay/240104004030

```
void setup() {
  Serial.begin(9600); // Start serial communication at 9600 bps
  pinMode(LED_BUILTIN, OUTPUT);
  char hello[] = "Hello, World!";
  int len = strlen(hello);
  for (int i = 0; i < len; i++) {
    blink(to_morse(toupper(hello[i]))); // Displays "Hello, World" during the setup, this is the first part of the assignment
  }
}
```

1. setup Function

The setup function, as the name suggests, sets sup the Arduino for input and output, it starts the communication and sets the output to the built in LED, and outputs “Hello, World!” in Morse code(first part of the assignment).

```
#define INTERVAL 250
```

2. #define

“#define” self explanatorily defines a constant called “INTERVAL” that can be set to any integer to determine duration of a dot, a dash is 3 times the interval and space is 7 times.

```
void loop() {
  if (Serial.available()) {
    char letter = Serial.read();  // Read the incoming letter
    blink(to_morse(toupper(Letter))); // Converts the character to dots and dashes and then to LED blinking, bonus part of the assignment
  }
}
```

3. loop Function

Bonus part of the assignment starts here, loop function checks for serial connection, if it is available, it reads the letters one by one and assigns it to a variable called “letter” of the type “char”. And then it feeds it to the “toupper”(which is a built-in) function that capitalizes the letter if it is lowercase.

```
// Converts characters to dots and dashes
const char* to_morse(char c) {
    switch (c) {
        // Letters A-Z
        case 'A': return ".-";
        case 'B': return "-...";
        case 'C': return "-.-.";
        ...
}
```

4. to_morse Function

The output of “toupper” then gets fed into the function called “to_morse” which translates the letter into Morse code using “.”, “-“ and “/”(for space). If it is a character that is not found e.g. Turkish letters(they have Morse codes but I did not implement them fort his project.), it is ignored.

```
// turns on and off the LED depending on the character
void blink(const char* code) {
  int len = strlen(code);  // Get the length of the Morse code string
  for (int i = 0; i < len; i++) {  // Iterates through each character in the string
    if (code[i] == '.') {  // Dot
      digitalWrite(LED_BUILTIN, HIGH);
      delay(INTERVAL);
      digitalWrite(LED_BUILTIN, LOW);
      delay(INTERVAL);
    }
    else if (code[i] == '-') {  // Dash
      digitalWrite(LED_BUILTIN, HIGH);
      delay(INTERVAL * 3);
      digitalWrite(LED_BUILTIN, LOW);
      delay(INTERVAL);
    }
    else if (code[i] == '/') {  // Space
      delay(INTERVAL * 7);
    }
  }
  delay(INTERVAL * 3);  // Gap between letters
}
```

5. blink Function

The output gets fed into another function called “blink”, it gets the length of the Morse code and iterates over every single character of the code and turns on and off the LED according to the character.

Note: “to_morse” and “blink” could’ve been implemented into a single function, translating the letters into dots and dashes is not necessary fort his project but it might be useful for unit testing or altering the code to make it output to a LCD screen etc.