Task 1.1: Run and connect an ESP32 to Wi-Fi.

Tools:

1. Esp32.
2. Arduino IDE.

Implementation steps:

1. Download and install Arduino IDE (Make sure to check the 'install USB driver' option when selecting the components to be installed in the Insulation Options step, to avoid any board recognition failure problems).
2. Setting up an ESP32 with Arduino IDE:
    a. The ESP32 board is not available if you use it for the first time with Arduino IDE, so you must navigate to File > Preferences and paste the following link in the Additional Boards Manager URL 'https://dl.espressif.com/dl/package_esp32_index.json'.
    b. After that navigate to Tools > Boards Manager and search for 'esp32' and install the library.
    c. The 'ESP32 Arduino' will appear in Tools > Boards "Arduino UNO", then select the "WEMOS D1 MINI ESP32" board.
3. To test if the board is working or not, we can run different examples that can be found from File > Example:
    a. First, plug the board into the USB driver using the micro-USB cable.
    b. Select the 'blink' example to test the board and plug the LED into the board or the 'get ship id' example if the LED is not available and press upload.
    c. If you chose the 'blink' example the LED will blink depending on the delay time you chose.
    d. If you chose the 'get ship id' example you can open the serial monitor to see the output (Make sure to select the same baud rate as the one in the setup() function in the code).
4. Moving to connect an ESP32 to WiFi.
    a. Include WiFi library at the top '#include <WiFi.h>'.
    b. Defined the name of the network and the password as constant variables.
    c. Inside the setup function or on aspirate function, defined the WiFi mode as 'WiFi.mode(WIFI_STA);' as we will connect to an existing WiFi network.

d. Make ESP32 connect to the WiFi network by calling ‘WiFi.begin()’ and passing the network name and password as parameters.

e. Next, create a while loop that keeps looping until we have a WiFi connection ‘while (WiFi.status() != WL_CONNECTED)’ inside the loop, we can print anything and it’s a good idea to define some delay time.

f. Finally, we can print to the serial monitor the WiFi local IP address.

```
#include <WiFi.h>

const char* WIFI_name = "";
const char* WIFI_password = "";

void setup() {
  // Connect to Wi-Fi network with WIFI name and password
  Serial.print("Connecting to ");
  Serial.println(WIFI_name);
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_name, WIFI_password);
  while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print("Connecting to WiFi ..");
  }
  // Print local IP address
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

}
```