Assignment P1

The first assignment is to implement a simple service for the serial interface, i.e., a module that accepts commands and responds accordingly. The device shall emit the string `SYNC␣//␣FIRMWARE␣READY` as soon as it has finished booting, i.e. when the `app_main(..)` has been entered. This is to simplify synchronization with other small serial programs running on the PC to which the device is attached.

At this stage, is shall operate as follows:

1. **MAC** command reports the device MAC address, e.g.:

   ```
   MAC␣AB:20:BC:30:DE:40
   ```

2. **ID** command reports the device ID in one line, which we define to be comma-separated list of the students' email addresses.

3. **DEC** command which converts a given numeric literal to decimal representation. More specifically, the input literal can be a binary, octal, hexadecimal or decimal number. The base is specified as in C source code, i.e., with prefixes `0b`, `0`, `0x`, respectively, whereas decimal numbers do not require any prefix.

   If the input is invalid (e.g., contains an invalid character as in `0xabch`, or the numerical value is larger than $2^{16} - 1$, then **ARGUMENT ERROR** response shall be transmitted.

4. **STATUS** command that reports the system uptime, the available cores, and the amount of free heap memory using the format below:

   ```
   SYSTEM␣UPTIME:␣12345␣S
   AVAILABLE␣CORES:␣2
   AVAILABLE␣HEAP␣MEMORY:␣12345
   ```

   - Uptime can be obtained using FreeRTOS system call `esp_timer_get_time(..)`, and S indicates the unit (seconds).
   - The number of cores is available by using ESP-specific system call `esp_chip_info(..)`
   - Available heap memory via `esp_get_free_heap_size(..)`

5. Any unknown command will be responded with a **COMMAND ERROR** line.

Note that:

- ␣ corresponds to a single space character
- Commands, arguments and responses are case insensitive.
- Server is suppose to send prompt, `>␣`, whenever it is ready to receive next command.
- Newlines are specified with a single character '\n'.

Example conversation, via serial_console.py :

```
...> python serial_console.py -D COM3

Console connected on serial port COM3
Reboot device to synchronize.
.............................................................
SYNC // FIRMWARE READY
> id
esa@hi.is, dff1@hi.is
> mac
MAC 24:0A:C4:61:04:00
> status
SYSTEM UPTIME: 29 s
AVAILABLE CORES: 2
AVAILABLE HEAP MEMORY: 300752
> dec 128
128
> dec 0x0080
128
> dec 0200
128
> dec 0b10000000
128
> dec 0xabcde
Argument error
> dec 0x000000abcd
43981
> dec 66000
Argument error
> dec 0b101020
Argument error
> dec 090
Argument error
> other_command
Command error
```

Useful Links

C standard library reference: `https://en.cppreference.com/w/c/header`

Espressif ESP32 API reference: `https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/index.html`

FreeRTOS Kernel API reference: `https://www.freertos.org/a00106.html`