



FreeRTOS on ESP32 Coding Challenge Problem

Your task is to design a program for the ESP32 using C++ and FreeRTOS that does the following.


- Ask through serial for a positive integer i , where $i \neq 0$ in char form ('123456').
- The program will count down from $n = 6$ once per second. Each time it does, it should:
 - If the current count n is even, a task on *Core 0* should output the string "Foo" to the serial monitor followed by the current number.
 - If the current count n is odd, then a task on *Core 1* should output the string "Bar" to the serial monitor followed by the current number.
 - If the number is prime, then a task on *Core 0* should output the string "Prime".
 - The expected sequence *output* for $i = 6$ should look something like this.

```
Received 6
Foo 6
Bar 5 Prime
Foo 4
Bar 3 Prime
Foo 2 Prime
Bar 1
Foo 0
```

- The next Foo (or Bar) must start outputting to the serial port one second after the previous Bar (or Foo) begins. In other words, the messages are exactly one second apart.
- At any time, the program may receive another integer.
 - If the program receives 0, then it should restart the ESP32 immediately.
 - You should acknowledge that you received the new number.
 - The program should finish the sequence it is currently doing first. In other words, it should reach "Foo 0" before starting the next number.
 - You may receive many numbers, do the sequences in the order you received them. If you receive more than 8 numbers while doing a sequence, you should ignore the 9th and output "Buffer is full".

You may assume that a valid value for i is given, $i < 2^{24}$, and your program runs on a standard ESP32-S3 chip.

Foo-Bar is a variation of the classic Fizz-Buzz test given in coding interviews; the basic functionality is easy to achieve, but a robust solution with good modularity and accurate timing will score better than a solution without these features.

 You can use simulators if you don't have access to an ESP32 module. The final code should be uploaded to a public Github repo.