

تمرین ۱:

با تغییر بخشی از کد به صورت زیر، یک Target برای مقدار نمایش عدد تعیین کرده و آن را اجرا می کنیم.

قطعه کد:

```
def timer_callback(self):  
    msg = String()  
    msg.data = '%d' % self.i  
  
    if self.i == 14:  
        self.timer.destroy()  
  
    self.publisher_.publish(msg)
```

خروجی:

```
kimia@kimia-VirtualBox:~/ws/src/p2/p2$ python3 main.py  
[INFO] [1677315513.616280246] [minimal_publisher]: Pub  
[INFO] [1677315514.092581567] [minimal_publisher]: Pub  
[INFO] [1677315514.590513728] [minimal_publisher]: Pub  
[INFO] [1677315515.093751431] [minimal_publisher]: Pub  
[INFO] [1677315515.590577392] [minimal_publisher]: Pub  
[INFO] [1677315516.111042058] [minimal_publisher]: Pub  
[INFO] [1677315516.596648131] [minimal_publisher]: Pub  
[INFO] [1677315517.090269206] [minimal_publisher]: Pub  
[INFO] [1677315517.590785762] [minimal_publisher]: Pub  
[INFO] [1677315518.091302787] [minimal_publisher]: Pub  
[INFO] [1677315518.590592855] [minimal_publisher]: Pub  
[INFO] [1677315519.091610945] [minimal_publisher]: Pub  
[INFO] [1677315519.590998018] [minimal_publisher]: Pub
```

تمرین ۲:

با تغییر بخشی از کد به صورت زیر، یک Target برای مقدار لاکپشت های کشته شده تعیین کرده و با هر بار کشته شدن هر کدام، یک Counter را آپدیت می کنیم. در نهایت کد را اجرا کرده و در ادامه نتیجه را می بینیم.

قطعه کد:

```
def __init__(self):
    super().__init__("spawner")
    self.x_ = 0
    self.y_ = 0
    self.name_ = "spawned"
    self.spawn_turtle()

    self.target = 14
    self.count = 0

    self.chaser_position_subscriber_ = self.create_subscription(Pose, "turtle1/

def callback_chased(self, chaser_position):
    if abs(chaser_position.x - self.x_) < 0.5 and abs(chaser_position.y - self.
        self.kill_turtle()
        time.sleep(0.1)
        self.spawn_turtle()
    else: pass

def kill_turtle(self):
    turtle_killer = self.create_client(Kill, "kill")

    turtle_to_kill = Kill.Request()
    turtle_to_kill.name = self.name_
```

خروجی:

