Homework 02/12/2023

Question 1 (20'):
Write a recursive function in python to calculate the 1+2+3+….+N, where N is the parameter of the function. Then, calculate 1+2+3+….+100=?

Question 2 (20'): Always moving straight

Write a python code to let your robot always move straightforward. That is, even though I manually change its direction during its moving, it always tries to keep moving forward along the original direction as if no interruptions. *Hint: You need to use the Gyro.*

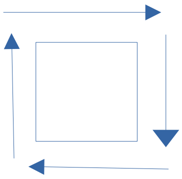
Note: In the class, all your driving base moves on a zig-zag trace. This is because you use the method start_tank(left_speed, right_speed). Can you find another way to let the driving base move more smooth?
Hint 1: Consider using proportional line following method.
Hint 2: You may need set_default_speed(speed) and start(steering=0, speed=None).

Question 3 (30'): Always move on the side of a square

Write a python code to let your robot keep moving on the edge of a square, no matter how it is interrupted during moving (i.e., its direction is changed slightly). The driving base will make a right turn every 4 seconds. It will terminate after moving 3 loops (i.e., stop when each side was passed three times)

Requirement: Please try defining a function, and then call this function in a *for* loop.

Hint: You need Timer, Gyro, etc.



Question 4 (30'): detect a small cargo vs. wall

Write a python function to check if a small cargo is in front of your driving base when it's moving. Specifically, let your robot move forward initially. If it encounters anything in front, it would try to determine whether it's a cargo or facing a wall.
The synopsis of the function is :
judge_cargo(motor_pair, distance_sensor)
If the object your robot encountered is really a cargo, return True; otherwise (it's facing a wall), return False.
***Hint**: motor_pair and distance_sensor are both instances/objects initialized outside the function. That is, they can be passed to the function and then used inside the function as usual.*
***Hint**: Here is an example to build a small cargo for testing your program.*
***Hint**: You can also extend the lifter by the black straight pole pieces in your spike prime box. Some exemplar images are here.*
***Hint**: If the distance_sensor faces an extremely long distance, it may return None value. That is, distance = None, but you can't use distance == None to make the judgement because None is not a number. This will throw an exception to crash the code.*
To judge this None value, you can use str(distance)=="None".