```
# Partie 1: Régulation en cap

import rospy
from turtlesim.msg import Pose

pose = Pose()  # Variable globale pour stocker la pose de la tortue

def pose_callback(data):
    global pose
    pose = data

def main():
    rospy.init_node('set_way_point_node')

    # Création du subscriber
    pose_sub = rospy.Subscriber('/turtle1/pose', Pose, pose_callback)

    # Définition du waypoint
    waypoint = Pose()
    waypoint.x = 7
    waypoint.y = 7

    rate = rospy.Rate(10)  # Fréquence de boucle (10 Hz)

    while not rospy.is_shutdown():
        # Calcul de l'angle désiré
        desired_angle = math.atan2(waypoint.y - pose.y, waypoint.x - pose.x)

        # Calcul de la commande en cap
        error = math.atan2(math.sin(desired_angle - pose.theta), math.cos(desired_angle - pose.theta))
        u = Kp * error
        # Publication de la commande en cap
        cmd_vel = Twist()
        cmd_vel.angular.z = u
        cmd_vel_pub.publish(cmd_vel)

        rate.sleep()

if __name__ == '__main__':
    try:
        main()
    except rospy.ROSInterruptException:
        pass
```