

```

"""
File that handles trig-math
"""
from math import sin, cos, pi, atan2, sqrt

def sign(a):
    """
    returns the sign of the float number
    """
    return int(a>0) - int(a<0)

def radToDeg(rad):
    """
    Convert from radians to degrees
    :param rad: radians
    :return: degrees
    """
    return rad * (180 /pi)

def degToRad(deg):
    """
    Convert from degrees to radians
    :param deg: degrees
    :return: radians
    """
    return deg * (pi /180)

def radToLaser(rad, robotDirection):
    """
    Takes an angle, the direction of the robot
    and returns the laser pointing closest to that position
    """
    return int(round(radToLaserFloat(rad,robotDirection)))

def radToLaserFloat(rad, robotDirection):
    """
    Takes an angle, the direction of the robot
    and returns a float that corresponds to the theoretical index
    that points in the correct direction.
    Example) 27.3 where index 27 or 28 would be the most suitable
    """
    rad -= robotDirection
    rad %= 2*pi
    if rad>pi: rad -= 2*pi
    #angle between -pi and pi
    laserAngle = radToDeg(rad)+135
    if laserAngle>270 or laserAngle<0: laserAngle=-1
    return laserAngle

def getCenterOfTurn(r, robotAngle, x, y):
    """
    If you have this radius and angle what is the center of the circle
    :param r: radius
    :param robotAngle: heading of the robot
    :param x: robots x-position
    :param y: robots y-position
    :return: the position of the center of circle of rotation
    """
    cx = x + r * cos(robotAngle + (pi / 2))
    cy = y + r * sin(robotAngle + (pi / 2))
    return cx, cy

def distanceToPoint(x,y,goalx,goaly):
    """
    The pythagorean distance from one point to another
    """
    return sqrt((goalx-x) ** 2 + (goaly-y) ** 2 )

```

```
def angleToPoint(x,y,goalx,goaly):
    """
        The angle of the vector between two points
        returns: radians
    """
    return atan2(goaly-y, goalx-x)

def angleDifference(angle,goalAngle):
    """
        The closest difference between two angles
        so 0,2*pi would return 0
        return: positive radians
    """
    a = angle - goalAngle
    a = ((a + pi) % (pi*2)) - pi
    return abs(a)

def angleDifferenceDirection(startAngle, endAngle,turnDir):
    """
        the difference between two angles of you follow the circle in a specified direction
        turnDir: 1 for counter-clockwise, -1 for clockwise
        returns: radians (non-absolute)
    """
    startAngle %= 2 * pi
    endAngle %= 2 * pi
    diff = (endAngle-startAngle) % (2 * pi)

    if turnDir==1: return diff
    else: return -((2*pi-diff) % (2*pi))

if __name__ == "__main__":
    print radToDeg(angleDifferenceDirection(degToRad(90),degToRad(0),-1))
```