```python
# NOTE:
#   Always import as:
#   from rotations import *

def make_Rbf(angles):
    from math import cos, sin, pi
    [psi, theta, phi] = angles

    psi = psi * (pi/180)
    theta = theta * (pi/180)
    phi = phi * (pi/180)

    R_bf = [[cos(psi)*cos(theta), sin(psi)*cos(theta), -sin(theta)],
            [cos(psi)*sin(theta)*sin(phi)-sin(psi)*cos(phi),
            cos(psi)*cos(phi)+sin(theta)*sin(psi)*sin(phi),
            cos(theta)*sin(phi)],
            [cos(psi)*sin(theta)*sin(phi)+sin(phi)*sin(psi),
            sin(theta)*sin(psi)*cos(phi)-sin(phi)*cos(psi),
            cos(theta)*cos(phi)]]

    return R_bf

def make_Rbw():
    from math import cos, sin, pi
    [alpha, beta] = angles

    # ***WARNING!***
    #If alpha or beta> 1 radian (60 deg), this breaks
    if alpha > 1 or beta > 1:
        alpha = alpha * (pi/180)
        beta = beta * (pi/180)

    R_bw = [[cos(alpha)*cos(beta), 0, 0],
            [sin(beta), 0, 0],
            [sin(alpha)*cos(beta), 0, 0]]

    return R_bw

def b2f(angles, vector):
    from numpy import matmul, transpose
    R_fb = transpose(make_Rbf(angles))
    return matmul(R_fb, vector)

def f2b(angles, vector):
    from numpy import matmul
    from numpy import matmul
    R_bf = make_Rbf(angles)
    return matmul(R_bf, vector)

def b2w(angles, vector):
    from numpy import matmul
    R_wb = transpose(make_Rbw(angles))
    return matmul(R_wb, vector)

def w2b(angles, vector):
    from numpy import matmul, transpose
    R_bw = make_Rbw(angles)
    return matmul(R_bw, vector)

```