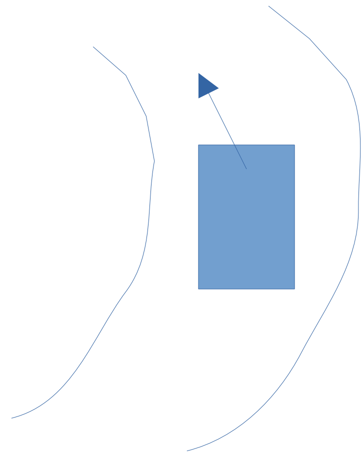
1.

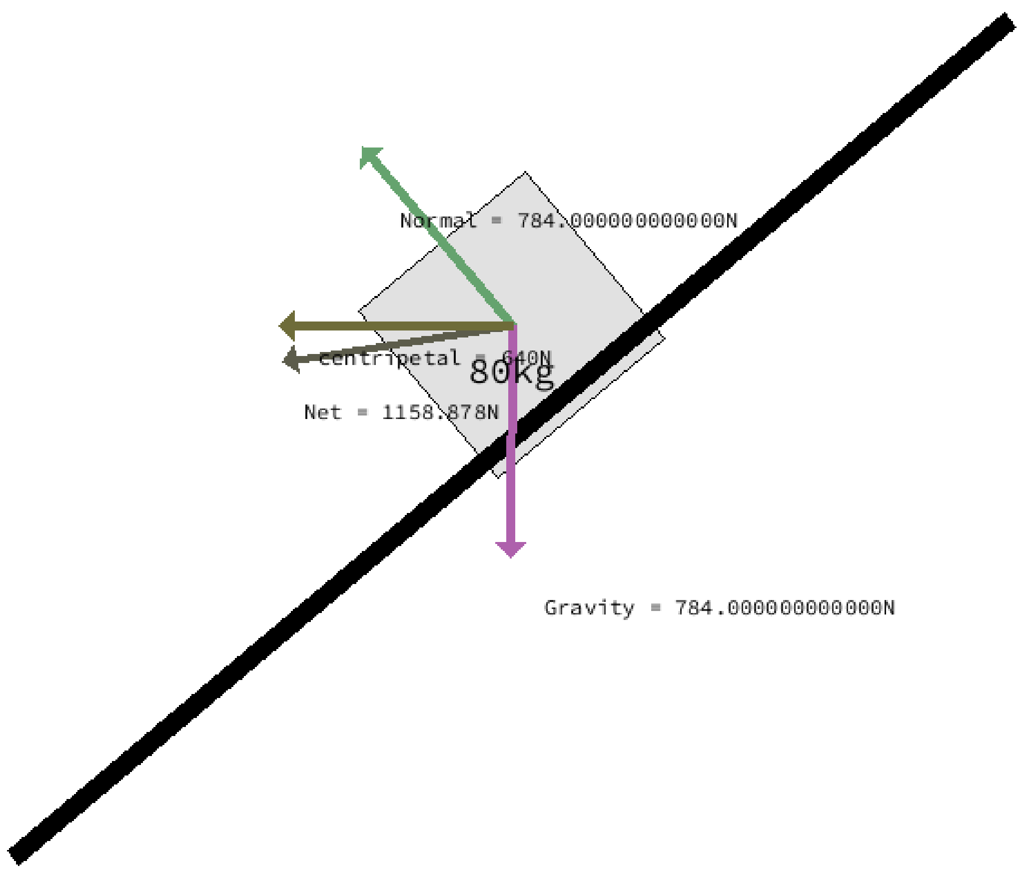


car = Particle2D(80, includeNormal=True, incline=rad(40))

car.addForce(Vector2D(newtonsLaw(a=centripitalMotion(v=40, r=200), m=80), 180, False), 'centripetal')

car.diagram()

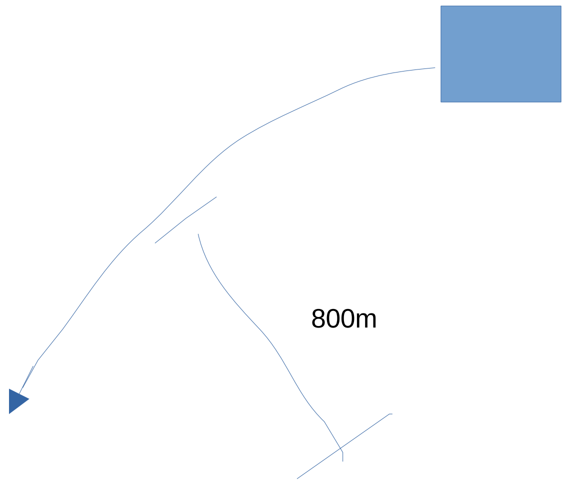
```
Net = 1158.878N @ -170.893°
Gravity = 784.000000000000N @ 270.0°
Normal = 784.000000000000N @ 130°
centripetal = 640N @ 180°
```



So,,, I'm not sure here… I guess b?
ANSWER: B

2.
mass = newtonsLawWeight(w=7*(10**2), a=9.8)
newtonsLaw(a=centripitalMotion(r=8*(10**2), v=2*(10**2)), m=mass)

ANSWER: a



3.

newtonsLaw(a=centripitalMotion(r=10, v=5), m=newtonsLawWeight(w=5000, a=9.8))

→ 1275.51020408163

Hey wait… What is j??

ANSWER: b

4. rps2angV just returns the angular velocity in radians/second from revolutions /second (it just multiplies it by 2pi)
r=10
v = swapAngLinearSpeed(vang=rps2angV(.1), r=r)
centripitalMotion(r=r, v=v).evalf()

→ 3.94784176043574

Which isn't even close to any of the answers, which means I have no idea how to do this problem.

5. Well not a or d…

And it’s not c cause the centrifugal force is blacklisted…

ANSWER: b

6. Um… don’t tilt? I’m still confused as to why you tilt axes, that just seems needlessly confusing.

That’s just something I need to ask someone about, I guess.

I guess c then?

ANSWER: c

7. centripitalMotion(a=9.8, v=100)

→ 1020.40816326531

ANSWER: b

8. This is what I got:

times = MappingList(2, 3, 4)

angles = MappingList([3-2*t**3 for t in times])

speeds = angles / times

→[-0.15384615384615385, -0.058823529411764705, -0.032]

Those are the velocites at times 2 seconds, 3 seconds, and 4 seconds.

(speeds[0] - speeds[1]) / 1

→ -0.09502262443438915

That *seems* right… ds/dt, right?

ANSWER: a?

9. Um… no?

angularAccel(angAccel=2, dt=2).evalf()

→ 4.0

That’s not 2.4 * 2*pi != 4. You’ve given me 3 values and they don’t work together. I must be missing something obvious here.

I guess if the wheel was already spinning and accelerating over a given 2 second interval, it could take 2.4 revolutions and still be accelerating at 2rad/s^2, but I have no idea how to translate that into math.

I guess, startingVelocity – endingVelocity = angAccel * dt? I think that makes sense?

But then I have 2 unknowns, and I can’t solve for that.

But I also have revolutions, so… lets see…

Still 9.

(2*pi*2.4)/2 = startingSpeed I think?

So…

solve(Eq((2*pi*2.4)/2 - var('x'), 2 * 2), x)

→ 3.53982236861550

Which is an answer! Cool!

ANSWER: e

10. So I want to find dAng

The only equation I have that involves dang is, angularSpeed

It then needs angular speed, and dt

I started doing this:

angularSpeed(angs=20, dt=5) + (angularAccel(dangs=20, dt=5) * ?)

Then I realized I can just do this:

motionEqu2(initp=0, initv=20, t=5, a=angularAccel(dangs=20, dt=5))/2*pi

→ 78.5398163397448

(motionEqu2 is $p = -\frac{at^2}{2} + initp + initvt$)

Which isn't an answer, so I don't know.

ANSWER: d??