Name: ____ Score: ____ / 100

There are 110 points possible.

1. (**21 points, 3 each**) In this question, you are given the following quantities. The question will ask you to show the numeric result of carrying out the given operation. For each, go as far as you can without a calculator. In general, I'm more interested in how you set up the operation, rather than the actual result (so you may want to show your work…).

$$M = \begin{bmatrix} 2 & 0 \\ 1 & 4 \\ 3 & -1 \end{bmatrix} \quad N = \begin{bmatrix} 5 & 0 & 9 \\ 8 & -6 & 3 \end{bmatrix} \quad \vec{v} = [4 \quad 5 \quad 1] \quad \vec{w} = [2 \quad 6 \quad 0]$$

   a. $\vec{v} + \vec{w}$

   b. $\vec{v} * \vec{w}$

   c. $\vec{v}$ x $\vec{w}$

   d. $\vec{v} \bullet \vec{w}$

   e. M * N

   f. $\vec{v}$ * M

   g. N * $\vec{w}$

2. (**10 points**) Write a python class called **Wave** that behaves as in the following program. Your job is to write the class so it will produce the given output (shown in comments). In python, all string objects have an upper and lower method (no parameters) that returns a upper-case or lower-case string, respectively (neither changes the original string).

```
X = Wave("bOb")                                    # Hi!
Y = Wave("wxYZ")                                   # Hi!
print(X)                                           # {{bOb}}
print(X[0])                                        # bob
print(Y[0])                                        # wxyz
print(X[1])                                        # BOB
print(Y[1])                                        # WXYZ
print(X + Y)                                       # Hi!
                                                    # {{bObwxYZ}}
Z = Y + X                                          # Hi!
Print(Z[0])                                        # wxyzbob
```

3. (**10 points**) Suppose you are given three lists (of equal size):
   a. **`Positions`**: A list of VectorN's (of dimension 2) representing the center of the spheres.
   b. **`Velocity`**: A list of VectorN's indicating the velocity of the spheres. The units are pixels / s.
   c. **`Radii`**: A list of integers indicating the radius of the spheres.

   For any index i (where i is between 0 and len(Positions)-1), Positions[i], Velocity[i] and Radii[i] are all the relevant values for sphere#i.

   Write python code to apply the acceleration due to gravity (stored in a global variable **G**, which is a VectorN) and bounce off the edges of the screen (in an 800 x 600 window). You can assume a variable **`dt`** (in seconds) has already been created and move the object (using the "integration" method we discussed in class) and then draw the circle to **`screen`** (again, already defined).

4. (**9 points**) Suppose you are given:
   a. $\vec{P}$: The position of the spaceship (in 3d) (expressed in feet from the center of the universe).
   b. $\hat{F}$ and $\hat{R}$: the forward and right direction of the spaceship (relative to the pilot) in a right-handed system.
   c. $\vec{E}$: the position of an enemy starfighter (expressed in feet from the center of the universe).

   Show a symbolic algorithm to determine the number of feet to the front, right, *and up* the enemy starship is relative to the pilot's frame of reference. For example the enemy might be 50 feet forward, -10 feet right (indicating the enemy is 10 feet to our left) and 32 feet up.

5. (**10 points**) Suppose you are given $\vec{a}$, $\vec{b}$, and $\vec{c}$ which make a triangle in 3d (assume none of them are equal). Show how to calculate the area of the triangle and the plane on which the triangle lies.

6. (**9 points**) If you are given:
   a. (ix, iy) the position of a pygame pixel on a (w x h) pygame surface.
   b. The camera's coordinate space (part of this question is knowing what you're given here)
   c. $\overrightarrow{vpo}$: The virtual view plane origin
   d. vpw, vph: the viewplane width and height (respectively) in world units
   e. screenWidth, screenHeight: the dimensions of the pygame window.

   Show how to calculate the 3d position of the pixel in our virtual world.

7. (**11 points**) If you are given:
    a. $\vec{C}$: the camera's position
    b. $\vec{P}$: the point we're illuminating on a sphere
    c. $\vec{X}$: the center of that sphere.
    d. $\overrightarrow{L_1}$: The position of Light #1
    e. $\overrightarrow{L_2}$: The position of Light #2
    f. $\overrightarrow{SL_1}$: The specular color of Light #1
    g. $\overrightarrow{SL_2}$: The specular color of Light #2
    h. $\overrightarrow{SM}$: The specular material of the sphere
    i. shininess: The specular shininess (some places in the lab / lecture this was called hardness – they mean the same thing).

    Show how to calculate the specular illumination at point $\vec{P}$.

8. (**9 points**) If you are given:
    a. $\vec{H}$: The position of a honey-bee (in 3d, expressed in feet)
    b. $\vec{F}$: The position of a flower (in 3d, expressed in feet)

    Show how to calculate the position of the honeybee if it moves 10 feet towards the flower. Also show (with python code) how we would calculate the angle the honeybee is facing while making that movement (assume we are viewing the bee from the +y direction, straight down in a right-handed coordinate system, with +x pointing to our right).

9. (**9 points**) If you are given:
    a. $\vec{G}$: The position of a patrolling guard (in 3d)
    b. $\vec{P}$: The position of the player (also in 3d)
    c. $\widehat{U}$: The direction the guard is facing (in 3d)
    d. q: $\cos(\theta)$, where θ describes the "cone-of-vision" of the guard.
    e. n: The maximum distance the guard can see.

    Write a symbolic algorithm to determine (True / False) if the guard can see the player. To get full points, don't use any trig functions.

10. (**9 points**) You are given:
    a. $\vec{T}$: The position of a tower (in 3d)
    b. L: A list of enemy positions (each element is a position in 3d)
    c. $\widehat{U}$: The current direction the tower turret is facing.

    Write a symbolic algorithm to determine the closest enemy and also the direction (clockwise / counter-clockwise) we should rotate in a left-handed system. You can assume the tower and all enemies are on the xz plane. To get full points, don't use any trig functions.

11. (**3 points**) This semester, I believe I deserve an ________ in the class (I'm just curious – there's no wrong answer here).