Q1

def find\_matrix\_shape(matrix):  
 return (len(matrix), len(matrix[1])) if matrix else (0, 0)

Q2

import numpy as np  
  
def compute\_cross\_product(array1, array2):  
 return np.cross(array1, array2)

Q3

import numpy as np  
  
def reconstruct\_matrix(U, S, V):  
 return np.dot(U, np.dot(np.diag(S), V))

Q4

A

import pygame  
from pygame.locals import \*  
  
def main():  
 pygame.init()  
 screen = pygame.display.set\_mode((500, 400))  
 pygame.display.set\_caption("My Canvas")  
 screen.fill((255, 255, 255)) # Fill the canvas with white color  
  
 running = True  
 while running:  
 for event in pygame.event.get():  
 if event.type == QUIT:  
 running = False  
 elif event.type == KEYDOWN and event.key == K\_F1:  
 running = False  
  
 pygame.display.flip()  
  
 pygame.quit()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

B

import pygame  
from pygame.locals import \*  
  
def main():  
 pygame.init()  
 screen = pygame.display.set\_mode((500, 400))  
 pygame.display.set\_caption("My Canvas")  
 screen.fill((255, 255, 255)) # Fill the canvas with white color  
  
 pygame.draw.line(screen, (255, 0, 0), (50, 50), (250, 50), 3) # Draw a red line  
  
 running = True  
 while running:  
 for event in pygame.event.get():  
 if event.type == QUIT:  
 running = False  
 elif event.type == KEYDOWN and event.key == K\_F1:  
 running = False  
  
 pygame.display.flip()  
  
 pygame.quit()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

C

import pygame  
from pygame.locals import \*  
  
def main():  
 pygame.init()  
 screen = pygame.display.set\_mode((500, 400))  
 pygame.display.set\_caption("My Canvas")  
 screen.fill((255, 255, 255)) # Fill the canvas with white color  
  
 # Draw triangles  
 pygame.draw.polygon(screen, (0, 0, 0), [(100, 100), (150, 50), (200, 100)])  
 pygame.draw.polygon(screen, (0, 0, 0), [(100, 200), (150, 150), (200, 200)])  
  
 running = True  
 while running:  
 for event in pygame.event.get():  
 if event.type == QUIT:  
 running = False  
 elif event.type == KEYDOWN and event.key == K\_F1:  
 running = False  
  
 pygame.display.flip()  
  
 pygame.quit()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

D

import pygame  
from pygame.locals import \*  
  
def main():  
 pygame.init()  
 screen = pygame.display.set\_mode((500, 400))  
 pygame.display.set\_caption("My Canvas")  
 screen.fill((255, 255, 255)) # Fill the canvas with white color  
  
 # Draw triangles  
 pygame.draw.polygon(screen, (0, 0, 0), [(100, 100), (150, 50), (200, 100)])  
 pygame.draw.polygon(screen, (0, 0, 0), [(100, 200), (150, 150), (200, 200)])  
  
 # Draw a purple point in the middle  
 pygame.draw.circle(screen, (128, 0, 128), (150, 125), 5)  
  
 running = True  
 while running:  
 for event in pygame.event.get():  
 if event.type == QUIT:  
 running = False  
 elif event.type == KEYDOWN and event.key == K\_F1:  
 running = False  
  
 pygame.display.flip()  
  
 pygame.quit()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Q5

import pygame  
from pygame.locals import \*  
from OpenGL.GL import \*  
from OpenGL.GLU import \*  
  
def draw\_triangle():  
 glBegin(GL\_TRIANGLES)  
 glColor3f(0.5, 0, 0.5) # Purple color  
 glVertex2f(-0.5, -0.5)  
 glVertex2f(0.5, -0.5)  
 glVertex2f(0, 0.5)  
 glEnd()  
  
def main():  
 pygame.init()  
 display = (800, 600)  
 pygame.display.set\_mode(display, DOUBLEBUF | OPENGL)  
 gluOrtho2D(-1, 1, -1, 1)  
  
 running = True  
 while running:  
 for event in pygame.event.get():  
 if event.type == QUIT:  
 running = False  
  
 glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)  
 draw\_triangle()  
 pygame.display.flip()  
 pygame.time.wait(10)  
  
 pygame.quit()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Q6

import pygame  
from pygame.locals import \*  
from OpenGL.GL import \*  
from OpenGL.GLU import \*  
  
  
def draw\_line(x1, y1, x2, y2):  
 glBegin(GL\_LINES)  
 glColor3f(1.0, 0.0, 0.0) # Red color  
 glVertex2f(x1, y1)  
 glVertex2f(x2, y2)  
 glEnd()  
  
  
def main():  
 pygame.init()  
 display = (800, 600)  
 pygame.display.set\_mode(display, DOUBLEBUF | OPENGL)  
 gluOrtho2D(-1, 1, -1, 1)  
  
 running = True  
 while running:  
 for event in pygame.event.get():  
 if event.type == QUIT:  
 running = False  
  
 glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)  
 draw\_line(-0.5, -0.5, 0.5, 0.5)  
 pygame.display.flip()  
 pygame.time.wait(10)  
  
 pygame.quit()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()