Task - 12 Simulate Garning concept using Vin: To simulate Garning concept using Pygame Snake game Write a python program to so create a snake Game wing pygame package. Algorithm 1 Import pygame package and initialize it 3. Create a snake class which initializes the snake 2. Define the window size Position and movement 4. create a function to check if the snake collides with the fruit 5. Create a game loop to continously update the game display. 6. End the game if the user quite Programs import pygame import time import scandom Brake\_ Speed = 15 Window\_ 2 = 720 Window-y = 480 black = py game. color (0,0,0) White = pygame. color (255, 255, 255) red: pygame. color (255,0,0) green = pygame. color (0, 255,0) blue = pygame. color (0,0,255)

Pyganse . ireit () Pygame. display. Set\_caption ('Geeks for Geek's snakes') garre - viredow = pygame . display . set - readle ((window-x, window) f Ps = pygame. time (lock() Snake - body = [[100, 50], [90, 50], [80,50], [70, 50] fruit - position = [varedom. vared range (1, (window\_x/10)) 10, random randrange (1, (window-y/10) 10] fruit - position: direction = (RIGHT' change - to = direction def show- score (choice, color, font, size): Score = 0 score - font = pygame. font. 848 font (font, 812e) Score - Surface - Score - font remder ('score: +8tr (score), true, color) Score - rect : Score - Sweface . get - rect () game - birdaw. blit (8 core - 8 wiface, 8 core - rect) def game = o ver (): my - fort = pygame. fort . Sys fort ('times new roman', 50) gome-over = my-font. vender ( 'Your Score is: 't str (Score), True, red) game\_over\_rect = game \_over - surface . get - reet () game - over - reet. midtop = (window - \*/2, window - 8/4) garre - window. blit (garre - over - & wefare, garre - over) Pygarre · display HIP() time. sleep (2) Py game. quit ()

```
quit ()
  for event in pygame. event. get ():
while true:
    if event type = = py games. Key Down:
       if event. Key == pygame . K - Up?
           charage - to = 'Up'
       if event. Key = = pygame . K. Down.
           change - to = Down
       if event. Key == Pygarra. K. Left:
           change - to = 'Left'
       if event . Key == Py games . K. Right:
           change - to = 'Right'
   if change - to = = 'up' and direction = Down':
           direction: Up'
   if change-to == 'Down' and direction! = 'up':
          derection: Down
   if change to = 'Left' and direction! = 'Right':
          direction: 'A Left':
   if change to: 'Right' and direction; = 'Left':
         direction : 'Right'
    if direction == 1 up':
          Snake-position[1]==10
    if direction = Down
          8 norke = -position[-] += 10
    if direction == 1 Left:
          8 nake - position [ ] - = 10
    if direction == 'Right':
           grake-position [0]+=10
    snake-booky. insert (o, list (snake-position))
   if snake-position [o] == fruit-position [o]
         = = fauit - position [i]:
          8 COTE + = 10
```

Birrielle by my concept warne To simulate General concept using Pagame , to a python program to see create a snother in a meing pagame package output: Troport pageme package and irution: 3703& Popure the window sixe create a make class which initializes the snake the tier and movement , seate a function to check if the snake collides with the senit a create a forme loop to contineously update the mane display. End the game of the used quits Pri yearn import organi estrit frogra mount som dom 1 reculte - Speed = 15 dindows 2 = 720 086 : 6-0-0 Darin 2.5 (45) reli - semant 1.0 - 11.00 (a . it - ) release a municipal war

else: Snake-body. Pop() if not fauit-spown: feuit - position = [randors . randrange (1, (window\_x/10)) \* 10, random. rand range (1, (window-yll10)) \*10] fait - Spawn = True game - window. gill (black) for pos in snake - body Pygame draw reet (game - window, green) Pygame reet (pos [0], Pos [J, 10,10]] Pygame draw reet (game - window, white, pygame reet) fruit-position [o], fruit position [i], 10,10) if snake-position[] < or snalre-position [] > window game - over () for block in snat. body [1:]: if snake-position [o] == black [o] and snake-position [i] = block [i]: game -over () 8 how - 8 core (1, white, 'times new roman', 20) Pygame display. update () fps. click (snake-speed) 2. Write a programe to Develop a chess board using py game. Program import pygame Pygame. init() screen - 812e = (640,640) Screen = Pygame. Display. Set\_mode (screen\_size)

```
black = (0,0,0)
White: (255, 255, 256)
brown = (153, 76,0)
def draw- board ():
   for you in range (8):
                                                Frankling
     for col in range (8):
        square - color = white if (vow +col) % 2 = = 0 else brown
        square - rect = Pygame . Rect (col 80, row 80, 80, 80, 80)
        Pygamo dua rect (screen square color, square-rect)
     del down-pieces (board):
      Piece - images = {
         'y': Pygame. image. Load ("mages /rook. Png").
         in : pygame image load ('image /knight. prog')-
         b': pygame image lood ('images/bishop. Png'),
          'q': pygame. image. load ('images/queen. Prg'),
          · K: Pygame · image · load ('images / King · Png'),
         p': Pygarre image load image/pour prog')
       For row in range (8):
          For col in range (8):
            Piece = board [row][col]
            if piece! - ":
               Piece-img = piece-images[piece]
              Piece * Tect = pygame . rect (col 80, row 80, 80, 80)
              screen blit (piece - image, piece-rect)
```

Les row is nonde (4):

output

Solver of (10) and of the state of 10) and of the state of 100 and of the state of 100 and of the state of 100 and of the state of the

( \* / 1550 ) ( \* / 1550 )

1. The series 6.

board = [ ['x','n','b','a',x',b','m','x'] ['p', 'p', 'p', 'p', 'p', 'p'], ['P', 'P', 'P', 'P', 'P', 'P'] ['R','N', B', Q', K', B', N', R'] graw-board () draw-pieces (boased) while True: for event in pygame. event. get (): of event. type == pygame. QUIT: Pygame quit () Py game. display update (1 Result: This the progame for pygames is executed and verified success fully.

	arch
	TECH 12
FORMANCE (5)	15 (5)
FINAL (3)	20
A. C.	13/10