```
#-----Statement of Authorship----------------------------------------#
#
#  This is an individual assessment item.  By submitting this
#  code I agree that it represents my own work.  I am aware of
#  the University rule that a student must not act in a manner
#  which constitutes academic dishonesty as stated and explained
#  in QUT's Manual of Policies and Procedures, Section C/5.3
#  "Academic Integrity" and Section E/2.1 "Student Code of Conduct".
#
#    Student no: n9983244
#    Student name: John Santias
#
#  NB: Files submitted without a completed copy of this statement
#  will not be marked.  All files submitted will be subjected to
#  software plagiarism analysis using the MoSS system
#  (http://theory.stanford.edu/~aiken/moss/).
#
#--------------------------------------------------------------------#



#-----Assignment Description-----------------------------------------#
#
#  BUILDING BLOCKS
#
#  This assignment tests your skills at defining functions, processing
#  data stored in lists and performing the arithmetic calculations
#  necessary to display a complex visual image.  The incomplete
#  Python script below is missing a crucial function, "stack_blocks".
#  You are required to complete this function so that when the
#  program is run it produces a picture of a pile of building blocks
#  whose arrangement is determined by data stored in a list which
#  specifies the blocks' locations.  See the instruction
#  sheet accompanying this file for full details.
#
#  Note that this assignment is in two parts, the second of which
#  will be released only just before the final deadline.  This
#  template file will be used for both parts and you will submit
#  your final solution as a single file, whether or not you
#  complete both parts of the assignment.
#
#--------------------------------------------------------------------#



#-----Preamble-------------------------------------------------------#
#
# This section imports necessary functions and defines constant
# values used for creating the drawing canvas.  You should not change
# any of the code in this section.
#

# Import the functions needed to complete this assignment.  You
# should not need to use any other modules for your solution.

from turtle import *
from math import *
```

```
# Define constant values used in the main program that sets up
# the drawing canvas.  Do not change any of these values.

block_size = 250 # pixels
top_and_bottom_border = 75 # pixels
left_and_right_border = 150 # pixels
canvas_width = (block_size + left_and_right_border) * 2
canvas_height = (block_size + top_and_bottom_border) * 2

#
#-----------------------------------------------------------------------#



#-----Functions for Managing the Canvas-----------------------------#
#
# The functions in this section are called by the main program to
# set up the drawing canvas for your image.  You should not change
# any of the code in this section.
#

# Set up the canvas and draw the background for the overall image
def create_drawing_canvas():

    # Set up the drawing canvas
    setup(canvas_width, canvas_height)

    # Set the coordinate system so that location (0, 0) is centred on
    # the point where the blocks will be stacked
    setworldcoordinates(-canvas_width / 2, -top_and_bottom_border,
                        canvas_width / 2, canvas_height -
top_and_bottom_border)

    # Draw as fast as possible
    tracer(False)

    # Colour the sky blue
    bgcolor('sky blue')

    # Draw the ground as a big green rectangle (sticking out of the
    # bottom edge of the drawing canvas slightly)
    overlap = 50 # pixels
    penup()
    goto(-(canvas_width / 2 + overlap), -(top_and_bottom_border +
overlap)) # start at the bottom-left
    fillcolor('pale green')
    begin_fill()
    setheading(90) # face north
    forward(top_and_bottom_border + overlap)
    right(90) # face east
    forward(canvas_width + overlap * 2)
    right(90) # face south
    forward(top_and_bottom_border + overlap)
    end_fill()
    penup()

    # Draw a friendly sun peeking into the image
```

```
    goto(-canvas_width / 2, block_size * 2)
    color('yellow')
    dot(250)

    # Reset everything ready for the student's solution
    color('black')
    width(1)
    penup()
    home()
    setheading(0)
    tracer(True)


# As a debugging aid, mark the coordinates of the centres and corners
# of the places where the blocks will appear
def mark_coords(show_corners = False, show_centres = False):

    # Go to each coordinate, draw a dot and print the coordinate, in the
given colour
    def draw_each_coordinate(colour):
        color(colour)
        for x_coord, y_coord in coordinates:
            goto(x_coord, y_coord)
            dot(4)
            write(' ' + str(x_coord) + ', ' + str(y_coord), font =
('Arial', 12, 'normal'))

    # Don't draw lines between the coordinates
    penup()

    # The list of coordinates to display
    coordinates = []

    # Only mark the corners if the corresponding argument is True
    if show_corners:
        coordinates = [[-block_size, block_size * 2], [0, block_size *
2], [block_size, block_size * 2],
                       [-block_size, block_size], [0, block_size],
[block_size, block_size],
                       [-block_size, 0], [0, 0], [block_size, 0]]
        draw_each_coordinate('dark blue')

    # Only mark the centres if the corresponding argument is True
    if show_centres:
        coordinates = [[-block_size / 2, block_size / 2], [block_size /
2, block_size / 2],
                       [-block_size / 2, block_size + block_size / 2],
[block_size / 2, block_size + block_size / 2]]
        draw_each_coordinate('red')

    # Put the cursor back how it was
    color('black')
    home()


# End the program by hiding the cursor and releasing the window
def release_drawing_canvas():
    tracer(True)
```

```
    hideturtle()
    done()

#
#-----------------------------------------------------------------------#



#-----Test data---------------------------------------------------------#
#
# These are the data sets you will use to test your code.
# Each of the data sets is a list specifying the locations of
# the building blocks:
#
# 1. The name of the block, from 'Block A' to 'Block D'
# 2. The place to put the block, either 'Top left', 'Top right',
#    'Bottom left' or 'Bottom right'
# 3. The block's orientation, meaning the direction in which the top
#    of the block is pointing, either 'Up', 'Down', 'Left' or 'Right'
# 4. An optional mystery value, 'X' or 'O', whose purpose will be
#    revealed only in the second part of the assignment
#
# Each data set does not necessarily mention all four blocks.
#
# You can create further data sets, but do not change any of the
# given ones below because they will be used to test your submission.
#

# The following data set doesn't require drawing any blocks
# at all.  You may find it useful as a dummy argument when you
# first start developing your "draw_attempt" function.

arrangement_00 = []

# Each of the following data sets specifies drawing just one block
# in an upright orientation.  You may find them useful when
# creating your individual pieces.

arrangement_01 = [['Block A', 'Bottom left', 'Up', 'O']]
arrangement_02 = [['Block B', 'Bottom right', 'Up', 'O']]
arrangement_03 = [['Block C', 'Bottom left', 'Up', 'O']]
arrangement_04 = [['Block D', 'Bottom right', 'Up', 'O']]

# Each of the following data sets specifies drawing just one block
# in non-upright orientations.  You may find them useful when
# ensuring that you can draw all the blocks facing in different
# directions.

arrangement_10 = [['Block A', 'Bottom left', 'Down', 'O']]
arrangement_11 = [['Block A', 'Bottom right', 'Left', 'O']]
arrangement_12 = [['Block A', 'Bottom left', 'Right', 'O']]

arrangement_13 = [['Block B', 'Bottom left', 'Down', 'O']]
arrangement_14 = [['Block B', 'Bottom right', 'Left', 'O']]
arrangement_15 = [['Block B', 'Bottom left', 'Right', 'O']]

arrangement_16 = [['Block C', 'Bottom left', 'Down', 'O']]
arrangement_17 = [['Block C', 'Bottom right', 'Left', 'O']]
```

```
arrangement_18 = [['Block C', 'Bottom left', 'Right', 'O']]

arrangement_19 = [['Block D', 'Bottom left', 'Down', 'O']]
arrangement_20 = [['Block D', 'Bottom right', 'Left', 'O']]
arrangement_21 = [['Block D', 'Bottom left', 'Right', 'O']]

# The following data sets all stack various numbers of
# blocks in various orientations

arrangement_30 = [['Block C', 'Bottom right', 'Up', 'O'],
                  ['Block D', 'Top right', 'Up', 'O']]

arrangement_32 = [['Block B', 'Bottom right', 'Up', 'O'],
                  ['Block D', 'Bottom left', 'Up', 'O'],
                  ['Block C', 'Top right', 'Up', 'O']]
arrangement_33 = [['Block B', 'Bottom right', 'Up', 'O'],
                  ['Block D', 'Bottom left', 'Up', 'O'],
                  ['Block A', 'Top left', 'Up', 'O']]
arrangement_34 = [['Block B', 'Bottom left', 'Up', 'O'],
                  ['Block A', 'Bottom right', 'Up', 'O'],
                  ['Block D', 'Top left', 'Up', 'O'],
                  ['Block C', 'Top right', 'Up', 'O']]

arrangement_40 = [['Block C', 'Bottom right', 'Left', 'O'],
                  ['Block D', 'Top right', 'Right', 'O']]

arrangement_42 = [['Block B', 'Bottom right', 'Left', 'O'],
                  ['Block D', 'Bottom left', 'Left', 'O'],
                  ['Block C', 'Top right', 'Down', 'O']]
arrangement_43 = [['Block B', 'Bottom right', 'Right', 'O'],
                  ['Block D', 'Bottom left', 'Left', 'O'],
                  ['Block A', 'Top left', 'Right', 'O']]
arrangement_44 = [['Block B', 'Bottom left', 'Down', 'O'],
                  ['Block A', 'Bottom right', 'Left', 'O'],
                  ['Block D', 'Top left', 'Right', 'O'],
                  ['Block C', 'Top right', 'Up', 'O']]

arrangement_50 = [['Block B', 'Bottom right', 'Left', 'O'],
                  ['Block D', 'Bottom left', 'Left', 'O'],
                  ['Block C', 'Top right', 'Down', 'X']]
arrangement_51 = [['Block B', 'Bottom right', 'Right', 'O'],
                  ['Block D', 'Bottom left', 'Left', 'O'],
                  ['Block A', 'Top left', 'Right', 'X']]
arrangement_52 = [['Block B', 'Bottom left', 'Down', 'O'],
                  ['Block A', 'Bottom right', 'Left', 'O'],
                  ['Block D', 'Top left', 'Right', 'O'],
                  ['Block C', 'Top right', 'Up', 'X']]

arrangement_60 = [['Block B', 'Bottom right', 'Left', 'X'],
                  ['Block D', 'Bottom left', 'Left', 'O'],
                  ['Block C', 'Top right', 'Down', 'O']]
arrangement_61 = [['Block B', 'Bottom right', 'Right', 'O'],
                  ['Block D', 'Bottom left', 'Left', 'X'],
                  ['Block A', 'Top left', 'Right', 'O']]
arrangement_62 = [['Block B', 'Bottom left', 'Down', 'X'],
                  ['Block A', 'Bottom right', 'Left', 'X'],
                  ['Block D', 'Top left', 'Right', 'O'],
                  ['Block C', 'Top right', 'Up', 'O']]
```

```
# The following arrangements create your complete image, but
# oriented the wrong way

arrangement_80 = [['Block C', 'Bottom right', 'Left', 'O'],
                  ['Block D', 'Top right', 'Left', 'X'],
                  ['Block A', 'Bottom left', 'Left', 'O'],
                  ['Block B', 'Top left', 'Left', 'O']]

arrangement_81 = [['Block B', 'Bottom right', 'Right', 'X'],
                  ['Block D', 'Bottom left', 'Right', 'X'],
                  ['Block A', 'Top right', 'Right', 'O'],
                  ['Block C', 'Top left', 'Right', 'O']]

arrangement_89 = [['Block A', 'Bottom right', 'Down', 'O'],
                  ['Block C', 'Top right', 'Down', 'O'],
                  ['Block B', 'Bottom left', 'Down', 'O'],
                  ['Block D', 'Top left', 'Down', 'O']]

# The following arrangements should create your complete image
# (but with the blocks stacked in a different order each time)

arrangement_90 = [['Block C', 'Bottom left', 'Up', 'O'],
                  ['Block D', 'Bottom right', 'Up', 'O'],
                  ['Block B', 'Top right', 'Up', 'X'],
                  ['Block A', 'Top left', 'Up', 'O']]

arrangement_91 = [['Block D', 'Bottom right', 'Up', 'X'],
                  ['Block C', 'Bottom left', 'Up', 'X'],
                  ['Block A', 'Top left', 'Up', 'O'],
                  ['Block B', 'Top right', 'Up', 'O']]

arrangement_92 = [['Block D', 'Bottom right', 'Up', 'X'],
                  ['Block B', 'Top right', 'Up', 'O'],
                  ['Block C', 'Bottom left', 'Up', 'O'],
                  ['Block A', 'Top left', 'Up', 'O']]

arrangement_99 = [['Block C', 'Bottom left', 'Up', 'O'],
                  ['Block D', 'Bottom right', 'Up', 'O'],
                  ['Block A', 'Top left', 'Up', 'O'],
                  ['Block B', 'Top right', 'Up', 'O']]

#
#-----------------------------------------------------------------------#



#-----Student's Solution-------------------------------------------------#
#
#  Complete the assignment by replacing the dummy function below with
#  your own "stack_blocks" function.
#


# Draw the stack of blocks as per the provided data set
def stack_blocks(arrangement):
```

```
#######-------------------------------- BLOCK A --------------------------
-----######
    def BlockA():

    #Draw block A's border
        pendown()
        begin_fill()
        width(2)
        color('white')
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        end_fill()
        penup()

    #Draw a quarter of a circle
        color('black')
        left(90)
        forward(250)
        left(90)
        begin_fill()
        pendown()
        circle(250, 90)
        left(90)
        forward(250)
        left(90)
        forward(250)
        end_fill()
        penup()

    #Draw red strip
        left(180)
        forward(45)
        right(90)
        pendown()
        begin_fill()
        color('#880019')
        how_long = range(15)
        for line_length in how_long:
            forward(21)
            left(6)
        forward(15)
        left(90)
        forward(50)
        left(90)
        how_long = range(19)
        for line_length in how_long:
              forward(13)
              right(5)
        left(5)
        forward(5)
        left(90)
        forward(50)
        end_fill()
```

```
        penup()

    #Draw top quarter of the letter "A"
        color('#fb1740')
        right(180)
        forward(11)
        right(25)
        pendown()
        begin_fill()
        forward(214)
        left(115)
        forward(64)
        left(65)
        forward(64)
        left(25)
        forward(135)
        end_fill()
        penup()

    #Seperate the A from the round strip
        pendown()
        forward(4)
        color('black')
        width(3)       #go up 2 extra pixels above the A
        left(155)
        forward(218)
        penup()

    #Overlay border
        left(115)
        forward(92)
        left(90)
        pendown()
        color('red')
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        penup()

######------------------------------ BLOCK B --------------------------
----######
    def BlockB():

    #Draw block B's border
         width(1)
         pendown()
         begin_fill()
         color('white')
         forward(249)
         left(90)
         forward(249)
         left(90)
         forward(249)
         left(90)
```

```
    forward(249)
    end_fill()
    penup()

#Draw quarter of a circle
    color('black')
    left(90)
    pendown()
    begin_fill()
    circle(249, 90)
    left(90)
    forward(249)
    left(90)
    forward(249)
    end_fill()
    penup()

 #Draw red strip
    left(180)
    forward(249)
    right(90)
    forward(205)
    right(90)
    pendown()
    begin_fill()
    color('#880019')
    how_long = range(15)
    for line_length in how_long:
        forward(21)
        right(6)
    forward(15)
    right(90)
    forward(55)
    right(90)
    how_long = range(19)
    for line_length in how_long:
          forward(13)
          left(5)
    right(5)
    right(90)
    forward(50)
    end_fill()
    penup()

 #Draw top quarter of letter A
    color('#fb1740')
    right(90)
    forward(69)
    right(90)
    forward(8)
    pendown()
    begin_fill()
    forward(197)
    right(90)
    forward(50)
    right(90)
    forward(101)
    left(155)
```

```
    forward(43)
    right(155) #look up
    forward(130)
    right(25)
    forward(15)
    right(65)
    how_long = range(7)
    for line_length in how_long:
      right(2)
      forward(9)
    end_fill()
    penup()

#Draw black lines to seperate the A
    color('black')
    left(14)
    forward(2)
    right(90)
    pendown()
    width(3)
    forward(195)
    penup()
    right(90)
    forward(71)
    right(90)
    forward(196)
    right(25)
    forward(2)
    pendown()
    forward(6)
    penup()
    left(115)
    forward(1)
    left(90)
    forward(174)
    left(45)
    width(3)
    forward(18)
    color('black')
    pendown()
    forward(24)
    penup()

 #Overlay border
    forward(3)
    right(135)
    color('red')
    forward(36)
    right(90)
    pendown()
    forward(250)
    right(90)
    forward(250)
    right(90)
    forward(250)
    right(90)
    forward(250)
    penup()
```

```
#######------------------------------ BLOCK C --------------------------
-----######

    def BlockC():

    #Draw block C's border
        width(1)
        pendown()
        begin_fill()
        color('white')
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        end_fill()
        penup()

    #Draw quarter of a circle
        color('black')
        begin_fill()
        left(90)
        forward(250)
        left(90)
        forward(250)
        left(90)
        pendown()
        circle(250, 90)
        end_fill()
        penup()


    #Draw the curved strip
        left(90)
        forward(45)
        color('#fb1740')
        left(90)
        pendown()
        begin_fill()
        color('#880019')
        how_long = range(15)
        for line_length in how_long:
            forward(21)
            right(6)
        forward(15)
        penup()
        right(90)
        forward(50)
        right(90)
        pendown()
        how_long = range(18)
        for line_length in how_long:
            forward(13)
            left(5)
        forward(17)
```

```
    right(90)
    forward(50)
    end_fill()
    penup()

#Draw the bottom quarter of the letter A
    left(180)
    forward(204)
    left(90)
    forward(91)
    color('#FB1740')
    left(65)
    pendown()
    begin_fill()
    forward(157)
    penup()
    right(180)
    forward(157)
    right(65)
    pendown()
    forward(66)
    left(245)
    forward(194)
    right(93)
    how_long = range(7)
    for line_length in how_long:
        right(2)
        forward(8)
    forward(2)
    end_fill()
    penup()

#Seperate the A from the round strip
    color('black')
    forward(4)
    width(3)
    right(73)
    pendown()
    forward(155)
    right(180)
    forward(160)
    left(180)
    forward(160)
    penup()
    right(65)
    forward(68)
    left(245)
    pendown()
    forward(197)
    penup()

#Draw horizontal support for A
    color('#FB1740')
    left(115)
    forward(106)
    left(90)
    forward(162)
    left(90)
```

```
        forward(37)
        begin_fill()
        pendown()
        right(180)
        forward(38)
        right(90)
        forward(50)
        right(90)
        forward(61)
        right(115)
        forward(53)
        end_fill()
        penup()

    #Overlay border
        right(65)
        forward(39)
        left(90)
        pendown()
        forward(19)
        left(90)
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        penup()

######------------------------------- BLOCK D --------------------------
----######

    def BlockD():

    #Draw block D's border
        width(1)
        pendown()
        begin_fill()
        color('white')
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        left(90)
        forward(250)
        end_fill()
        penup()

    #Draw quarter of a circle
        color('black')
        left(180)
        forward(250)
        begin_fill()
        right(180)
```

```
    pendown()
    circle(250, 90)
    left(90)
    forward(250)
    left(90)
    forward(250)
    end_fill()
    penup()

#Draw the curved strip
    left(180)
    forward(45)
    pendown()
    begin_fill()
    color('#880019')
    right(90)
    how_long = range(15)
    for line_length in how_long:
        forward(21)
        left(6)
    forward(15)
    left(90)
    forward(54)
    left(90)
    how_long = range(18)
    for line_length in how_long:
        forward(13)
        right(5)
    forward(14)
    left(90)
    forward(50)
    end_fill()
    penup()

#Draw the bottom quarter of A
    right(180)
    forward(190)
    color('#FB1740')
    begin_fill()
    right(90)
    pendown()
    forward(20)
    left(90)
    forward(15)
    right(90)
    forward(50)
    right(90)
    forward(100)
    right(90)
    forward(50)
    right(90)
    forward(34)
    left(90)
    forward(20)
    right(90)
    forward(50)
    end_fill()
    penup()
```

```
    #Draw black lines through A
        forward(15)
        right(90)
        forward(35)
        pendown()
        width(3)
        color('black')
        right(45)
        forward(50)
        right(90)
        forward(72)
        penup()

    #Overlay border
        right(45)
        forward(20)
        right(90)
        forward(88)
        right(90)
        color('red')
        pendown()
        forward(250)
        right(90)
        forward(250)
        right(90)
        forward(250)
        right(90)
        forward(250)
        penup()


##########---------------------------- IF STATEMENTS ----------------------
------##########

    if len(arrangement) == 0 :
        pass
##########------------------------- IF ARRANGEMENT[0] --------------------
------##########

######------ ARRANGEMENT[0] for BLOCK A ------######
    if len(arrangement) >= 1: #if the length of the arrangement list is
greater or less \
    #than 1
        if arrangement[0][0] == "Block A":
          if arrangement[0][2] == "Up":
            setheading(90)
            if arrangement[0][1] == "Bottom left":
                goto(0, 0)
                BlockA()
            elif arrangement[0][1] == "Top left":
                goto(0, 250)
                BlockA()
            elif arrangement[0][1] == "Bottom right":
                goto(250, 0)
                BlockA()
            elif arrangement[0][1] == "Top right":
                goto(250, 250)
```

```
                BlockA()

        elif arrangement[0][2] =="Down":
            setheading(270)
            if arrangement[0][1] == "Bottom left":
                goto(-250, 250)
                BlockA()
            elif arrangement[0][1] == "Top left":
                goto(-250, 500)
                BlockA()
            elif arrangement[0][1] == "Bottom right":
                goto(0, 250)
                BlockA()
            elif arrangement[0][1] == "Top right":
                goto(0, 500)
                BlockA()

        elif arrangement[0][2] == "Left":
            setheading(180)
            if arrangement[0][1] == "Bottom left":
                goto(0, 250)
                BlockA()
            elif arrangement[0][1]== "Top left":
                goto(250, 500)
                BlockA()
            elif arrangement[0][1]== "Bottom right":
                goto(250, 250)
                BlockA()
            elif arrangement[0][1] == "Top right":
                goto(250, 500)
                BlockA()

        elif arrangement[0][2] == "Right":
            setheading(0)
            if arrangement[0][1] == "Bottom left":
                goto(-250, 0)
                BlockA()
            elif arrangement[0][1]== "Top left":
                goto(-250, 250)
                BlockA()
            elif arrangement[0][1]== "Bottom right":
                goto(0, 0)
                BlockA()
            elif arrangement[0][1] == "Top right":
                goto(0, 250)
                BlockA()

######------ ARRANGEMENT[0] for BLOCK B ------######
        elif arrangement[0][0] == "Block B":
                if arrangement[0][2] == "Up":
                  setheading(90)
                  if arrangement[0][1] == "Bottom left":
                    goto(0, 0)
                    BlockB()
                  elif arrangement[0][1]== "Top left":
                    goto(0, 250)
                    BlockB()
                  elif arrangement[0][1]== "Bottom right":
```

```
                    goto(250, 0)
                    BlockB()
                  elif arrangement[0][1] == "Top right":
                    goto(250, 250)
                    BlockB()

                elif arrangement[0][2] == "Down":
                  setheading(270)
                  if arrangement[0][1] == "Bottom left":
                    goto(-250, 250)
                    BlockB()
                  elif arrangement[0][1]== "Top left":
                    goto(-250, 500)
                    BlockB()
                  elif arrangement[0][1]== "Bottom right":
                    goto(0, 250)
                    BlockB()
                  elif arrangement[0][1] == "Top right":
                    goto(0, 500)
                    BlockB()

                elif arrangement[0][2] == "Left":
                  setheading(180)
                  if arrangement[0][1] == "Bottom left":
                    goto(0, 250)
                    BlockB()
                  elif arrangement[0][1]== "Top left":
                    goto(0, 500)
                    BlockB()
                  elif arrangement[0][1]== "Bottom right":
                    goto(250, 250)
                    BlockB()
                  elif arrangement[0][1] == "Top right":
                    goto(250, 500)
                    BlockB()

                elif arrangement[0][2] == "Right":
                  setheading(0)
                  if arrangement[0][1] == "Bottom left":
                    goto(-250, 0)
                    BlockB()
                  elif arrangement[0][1]== "Top left":
                    goto(-250, 250)
                    BlockB()
                  elif arrangement[0][1]== "Bottom right":
                    goto(0, 0)
                    BlockB()
                  elif arrangement[0][1] == "Top right":
                    goto(0, 250)
                    BlockB()

######------ ARRANGEMENT[0] for BLOCK C ------######
        elif arrangement[0][0] == "Block C":
                if arrangement[0][2] == "Up":
                  setheading(90)
                  if arrangement[0][1] == "Bottom left":
                    goto(0, 0)
                    BlockC()
```

```
                elif arrangement[0][1] == "Top left":
                  goto(0, 250)
                  BlockC()
                elif arrangement[0][1]== "Bottom right":
                  goto(250, 0)
                  BlockC()
                elif arrangement[0][1]== "Top right":
                  goto(250, 250)
                  BlockC()

              elif arrangement[0][2] == "Down":
                setheading(270)
                if arrangement[0][1] == "Bottom left":
                  goto(-250, 250)
                  BlockC()
                elif arrangement[0][1] == "Top left":
                  goto(-250, 500)
                  BlockC()
                elif arrangement[0][1] == "Bottom right":
                  goto(0, 250)
                  BlockC()
                elif arrangement[0][1] == "Top right":
                  goto(0, 500)
                  BlockC()

              elif arrangement[0][2] == "Left":
                setheading(180)
                if arrangement[0][1] == "Bottom left":
                  goto(0, 250)
                  BlockC()
                elif arrangement[0][1] == "Top left":
                  goto(250, 500)
                  BlockC()
                elif arrangement[0][1] == "Bottom right":
                  goto(250, 250)
                  BlockC()
                elif arrangement[0][1] == "Top right":
                  goto(250, 500)
                  BlockC()

              elif arrangement[0][2] == "Right":
                setheading(0)
                if arrangement[0][1] == "Bottom left":
                  goto(-250, 0)
                  BlockC()
                elif arrangement[0][1] == "Top left":
                  goto(-250, 250)
                  BlockC()
                elif arrangement[0][1] == "Bottom right":
                  goto(0, 0)
                  BlockC()
                elif arrangement[0][1] == "Top right":
                  goto(0, 250)
                  BlockC()

######------ ARRANGEMENT[0] for BLOCK D ------######
        elif arrangement[0][0] == "Block D":
                if arrangement[0][2] == "Up":
```

```
  setheading(90)
  if arrangement[0][1] == "Bottom left":
    goto(0, 0)
    BlockD()
  elif arrangement[0][1]== "Top left":
    goto(0, 250)
    BlockD()
  elif arrangement[0][1]== "Bottom right":
    goto(250, 0)
    BlockD()
  elif arrangement[0][1] == "Top right":
    goto(250, 250)
    BlockD()

elif arrangement[0][2] == "Down":
  setheading(270)
  if arrangement[0][1] == "Bottom left":
    goto(-250, 250)
    BlockD()
  elif arrangement[0][1]== "Top left":
    goto(-250, 500)
    BlockD()
  elif arrangement[0][1]== "Bottom right":
    goto(0, 250)
    BlockD()
  elif arrangement[0][1] == "Top right":
    goto(0, 500)
    BlockD()

elif arrangement[0][2] == "Left":
  setheading(180)
  if arrangement[0][1] == "Bottom left":
    goto(0, 250)
    BlockD()
  elif arrangement[0][1]== "Top left":
    goto(0, 500)
    BlockD()
  elif arrangement[0][1]== "Bottom right":
    goto(250, 250)
    BlockD()
  elif arrangement[0][1] == "Top right":
    goto(250, 500)
    BlockD()

elif arrangement[0][2] == "Right":
  setheading(0)
  if arrangement[0][1] == "Bottom left":
    goto(-250, 0)
    BlockD()
  elif arrangement[0][1]== "Top left":
    goto(-250, 250)
    BlockD()
  elif arrangement[0][1]== "Bottom right":
    goto(0, 0)
    BlockD()
  elif arrangement[0][1] == "Top right":
    goto(0, 250)
    BlockD()
```

```
    else:
          pass

##########----------------------END OF  IF ARRANGEMENT[0] ---------------
------##########

##########------------------------- IF ARRANGEMENT[1] -------------------
------##########
######------ ARRANGEMENT[1] for BLOCK A ------######
    if len(arrangement) >= 2:
        if arrangement[1][0] == "Block A":
                if arrangement[1][2] == "Up":
                  setheading(90)
                  if arrangement[1][1] == "Bottom left":
                    goto(0, 0)
                    BlockA()
                  elif arrangement[1][1]== "Top left":
                    goto(0, 250)
                    BlockA()
                  elif arrangement[1][1]== "Bottom right":
                    goto(250, 0)
                    BlockA()
                  elif arrangement[1][1] == "Top right":
                    goto(250, 250)
                    BlockA()

                elif arrangement[1][2] == "Down":
                  setheading(270)
                  if arrangement[1][1] == "Bottom left":
                    goto(-250, 250)
                    BlockA()
                  elif arrangement[1][1]== "Top left":
                    goto(-250, 500)
                    BlockA()
                  elif arrangement[1][1]== "Bottom right":
                    goto(0, 250)
                    BlockA()
                  elif arrangement[1][1] == "Top right":
                    goto(0, 500)
                    BlockA()

                elif arrangement[1][2] == "Left":
                  setheading(180)
                  if arrangement[1][1] == "Bottom left":
                    goto(0, 250)
                    BlockA()
                  elif arrangement[1][1]== "Top left":
                    goto(250, 500)
                    BlockA()
                  elif arrangement[1][1]== "Bottom right":
                    goto(250, 250)
                    BlockA()
                  elif arrangement[1][1] == "Top right":
                    goto(250, 500)
                    BlockA()

                elif arrangement[1][2] == "Right":
```

```
                setheading(0)
                if arrangement[1][1] == "Bottom left":
                  goto(-250, 0)
                  BlockA()
                elif arrangement[1][1]== "Top left":
                  goto(-250, 250)
                  BlockA()
                elif arrangement[1][1]== "Bottom right":
                  goto(0, 0)
                  BlockA()
                elif arrangement[1][1] == "Top right":
                  goto(0, 250)
                  BlockA()

######------ ARRANGEMENT[1] for BLOCK B ------######
        elif arrangement[1][0] == "Block B":
                if arrangement[1][2] == "Up":
                  setheading(90)
                  if arrangement[1][1] == "Bottom left":
                    goto(0, 0)
                    BlockB()
                  elif arrangement[1][1]== "Top left":
                    goto(0, 250)
                    BlockB()
                  elif arrangement[1][1]== "Bottom right":
                    goto(250, 0)
                    BlockB()
                  elif arrangement[1][1] == "Top right":
                    goto(250, 250)
                    BlockB()

                elif arrangement[1][2] == "Down":
                  setheading(270)
                  if arrangement[1][1] == "Bottom left":
                    goto(-250, 250)
                    BlockB()
                  elif arrangement[1][1]== "Top left":
                    goto(-250, 500)
                    BlockB()
                  elif arrangement[1][1]== "Bottom right":
                    goto(0, 250)
                    BlockB()
                  elif arrangement[1][1] == "Top right":
                    goto(0, 500)
                    BlockB()

                elif arrangement[1][2] == "Left":
                  setheading(180)
                  if arrangement[1][1] == "Bottom left":
                    goto(0, 250)
                    BlockB()
                  elif arrangement[1][1]== "Top left":
                    goto(0, 500)
                    BlockB()
                  elif arrangement[1][1]== "Bottom right":
                    goto(250, 250)
                    BlockB()
                  elif arrangement[1][1] == "Top right":
```

```
                    goto(250, 500)
                    BlockB()

                elif arrangement[1][2] == "Right":
                  setheading(0)
                  if arrangement[1][1] == "Bottom left":
                    goto(-250, 0)
                    BlockB()
                  elif arrangement[1][1]== "Top left":
                    goto(-250, 250)
                    BlockB()
                  elif arrangement[1][1]== "Bottom right":
                    goto(0, 0)
                    BlockB()
                  elif arrangement[1][1] == "Top right":
                    goto(0, 250)
                    BlockB()

######------ ARRANGEMENT[1] for BLOCK C ------######
        elif arrangement[1][0] == "Block C":
                if arrangement[1][2] == "Up":
                  setheading(90)
                  if arrangement[1][1] == "Bottom left":
                    goto(0, 0)
                    BlockC()
                  elif arrangement[1][1]== "Top left":
                    goto(0, 250)
                    BlockC()
                  elif arrangement[1][1]== "Bottom right":
                    goto(250, 0)
                    BlockC()
                  elif arrangement[1][1] == "Top right":
                    goto(250, 250)
                    BlockC()

                elif arrangement[1][2] == "Down":
                  setheading(270)
                  if arrangement[1][1] == "Bottom left":
                    goto(-250, 250)
                    BlockC()
                  elif arrangement[1][1]== "Top left":
                    goto(-250, 500)
                    BlockC()
                  elif arrangement[1][1]== "Bottom right":
                    goto(0, 250)
                    BlockC()
                  elif arrangement[1][1] == "Top right":
                    goto(0, 500)
                    BlockC()

                elif arrangement[1][2] == "Left":
                  setheading(180)
                  if arrangement[1][1] == "Bottom left":
                    goto(0, 250)
                    BlockC()
                  elif arrangement[1][1]== "Top left":
                    goto(250, 500)
                    BlockC()
```

```
                elif arrangement[1][1]== "Bottom right":
                  goto(250, 250)
                  BlockC()
                elif arrangement[1][1] == "Top right":
                  goto(250, 500)
                  BlockC()

              elif arrangement[1][2] == "Right":
                setheading(0)
                if arrangement[1][1] == "Bottom left":
                  goto(-250, 0)
                  BlockC()
                elif arrangement[1][1]== "Top left":
                  goto(-250, 250)
                  BlockC()
                elif arrangement[1][1]== "Bottom right":
                  goto(0, 0)
                  BlockC()
                elif arrangement[1][1] == "Top right":
                  goto(0, 250)
                  BlockC()

######------ ARRANGEMENT[1] for BLOCK D ------######
        elif arrangement[1][0] == "Block D":
                if arrangement[1][2] == "Up":
                  setheading(90)
                  if arrangement[1][1] == "Bottom left":
                    goto(0, 0)
                    BlockD()
                  elif arrangement[1][1]== "Top left":
                    goto(0, 250)
                    BlockD()
                  elif arrangement[1][1]== "Bottom right":
                    goto(250, 0)
                    BlockD()
                  elif arrangement[1][1] == "Top right":
                    goto(250, 250)
                    BlockD()

                elif arrangement[1][2] == "Down":
                  setheading(270)
                  if arrangement[1][1] == "Bottom left":
                    goto(-250, 250)
                    BlockD()
                  elif arrangement[1][1]== "Top left":
                    goto(-250, 500)
                    BlockD()
                  elif arrangement[1][1]== "Bottom right":
                    goto(0, 250)
                    BlockD()
                  elif arrangement[1][1] == "Top right":
                    goto(0, 500)
                    BlockD()

                elif arrangement[1][2] == "Left":
                  setheading(180)
                  if arrangement[1][1] == "Bottom left":
                    goto(0, 250)
```

```
                    BlockD()
                  elif arrangement[1][1]== "Top left":
                    goto(0, 500)
                    BlockD()
                  elif arrangement[1][1]== "Bottom right":
                    goto(250, 250)
                    BlockD()
                  elif arrangement[1][1] == "Top right":
                    goto(250, 500)
                    BlockD()

                elif arrangement[1][2] == "Right":
                  setheading(0)
                  if arrangement[1][1] == "Bottom left":
                    goto(-250, 0)
                    BlockD()
                  elif arrangement[1][1]== "Top left":
                    goto(-250, 250)
                    BlockD()
                  elif arrangement[1][1]== "Bottom right":
                    goto(0, 0)
                    BlockD()
                  elif arrangement[1][1] == "Top right":
                    goto(0, 250)
                    BlockD()
    else:
         pass

##########----------------------END OF  IF ARRANGEMENT[1] ---------------
------##########

##########------------------------- IF ARRANGEMENT[2] -------------------
------##########
######------ ARRANGEMENT[2] for BLOCK A ------######
    if len(arrangement) >= 3:
        if arrangement[2][0] == "Block A":
                if arrangement[2][2] == "Up":
                  setheading(90)
                  if arrangement[2][1] == "Bottom left":
                    goto(0, 0)
                    BlockA()
                  elif arrangement[2][1]== "Top left":
                    goto(0, 250)
                    BlockA()
                  elif arrangement[2][1]== "Bottom right":
                    goto(250, 0)
                    BlockA()
                  elif arrangement[2][1] == "Top right":
                    goto(250, 250)
                    BlockA()

                elif arrangement[2][2] == "Down":
                  setheading(270)
                  if arrangement[2][1] == "Bottom left":
                    goto(-250, 250)
                    BlockA()
                  elif arrangement[2][1]== "Top left":
                    goto(-250, 500)
```

```
                BlockA()
              elif arrangement[2][1]== "Bottom right":
                goto(0, 250)
                BlockA()
              elif arrangement[2][1] == "Top right":
                goto(0, 500)
                BlockA()

            elif arrangement[2][2] == "Left":
              setheading(180)
              if arrangement[2][1] == "Bottom left":
                goto(0, 250)
                BlockA()
              elif arrangement[2][1]== "Top left":
                goto(250, 500)
                BlockA()
              elif arrangement[2][1]== "Bottom right":
                goto(250, 250)
                BlockA()
              elif arrangement[2][1] == "Top right":
                goto(250, 500)
                BlockA()

            elif arrangement[2][2] == "Right":
              setheading(0)
              if arrangement[2][1] == "Bottom left":
                goto(-250, 0)
                BlockA()
              elif arrangement[2][1]== "Top left":
                goto(-250, 250)
                BlockA()
              elif arrangement[2][1]== "Bottom right":
                goto(0, 0)
                BlockA()
              elif arrangement[2][1] == "Top right":
                goto(0, 250)

######------ ARRANGEMENT[2] for BLOCK B ------######
        elif arrangement[2][0] == "Block B":
                if arrangement[2][2] == "Up":
                  setheading(90)
                  if arrangement[2][1] == "Bottom left":
                    goto(0, 0)
                    BlockB()
                  elif arrangement[2][1]== "Top left":
                    goto(0, 250)
                    BlockB()
                  elif arrangement[2][1]== "Bottom right":
                    goto(250, 0)
                    BlockB()
                  elif arrangement[2][1] == "Top right":
                    goto(250, 250)
                    BlockB()

                elif arrangement[2][2] == "Down":
                  setheading(270)
                  if arrangement[2][1] == "Bottom left":
                    goto(-250, 250)
```

```
                BlockB()
              elif arrangement[2][1]== "Top left":
                goto(-250, 500)
                BlockB()
              elif arrangement[2][1]== "Bottom right":
                goto(0, 250)
                BlockB()
              elif arrangement[2][1] == "Top right":
                goto(0, 500)
                BlockB()

            elif arrangement[2][2] == "Left":
              setheading(180)
              if arrangement[2][1] == "Bottom left":
                goto(0, 250)
                BlockB()
              elif arrangement[2][1]== "Top left":
                goto(0, 500)
                BlockB()
              elif arrangement[2][1]== "Bottom right":
                goto(250, 250)
                BlockB()
              elif arrangement[2][1] == "Top right":
                goto(250, 500)
                BlockB()

            elif arrangement[2][2] == "Right":
              setheading(0)
              if arrangement[2][1] == "Bottom left":
                goto(-250, 0)
                BlockB()
              elif arrangement[2][1]== "Top left":
                goto(-250, 250)
                BlockB()
              elif arrangement[2][1]== "Bottom right":
                goto(0, 0)
                BlockB()
              elif arrangement[2][1] == "Top right":
                goto(0, 250)
                BlockB()

######------ ARRANGEMENT[2] for BLOCK C ------######
        elif arrangement[2][0] == "Block C":
                if arrangement[2][2] == "Up":
              setheading(90)
              if arrangement[2][1] == "Bottom left":
                goto(0, 0)
                BlockC()
              elif arrangement[2][1]== "Top left":
                goto(0, 250)
                BlockC()
              elif arrangement[2][1]== "Bottom right":
                goto(250, 0)
                BlockC()
              elif arrangement[2][1] == "Top right":
                goto(250, 250)
                BlockC()
```

```
            elif arrangement[2][2] == "Down":
              setheading(270)
              if arrangement[2][1] == "Bottom left":
                goto(-250, 250)
                BlockC()
              elif arrangement[2][1]== "Top left":
                goto(-250, 500)
                BlockC()
              elif arrangement[2][1]== "Bottom right":
                goto(0, 250)
                BlockC()
              elif arrangement[2][1] == "Top right":
                goto(0, 500)
                BlockC()

            elif arrangement[2][2] == "Left":
              setheading(180)
              if arrangement[2][1] == "Bottom left":
                goto(0, 250)
                BlockC()
              elif arrangement[2][1]== "Top left":
                goto(250, 500)
                BlockC()
              elif arrangement[2][1]== "Bottom right":
                goto(250, 250)
                BlockC()
              elif arrangement[2][1] == "Top right":
                goto(250, 500)
                BlockC()

            elif arrangement[2][2] == "Right":
              setheading(0)
              if arrangement[2][1] == "Bottom left":
                goto(-250, 0)
                BlockC()
              elif arrangement[2][1]== "Top left":
                goto(-250, 250)
                BlockC()
              elif arrangement[2][1]== "Bottom right":
                goto(0, 0)
                BlockC()
              elif arrangement[2][1] == "Top right":
                goto(0, 250)
                BlockC()


######------ ARRANGEMENT[2] for BLOCK D ------######
        elif arrangement[2][0] == "Block D":
              if arrangement[2][2] == "Up":
                setheading(90)
                if arrangement[2][1] == "Bottom left":
                  goto(0, 0)
                  BlockD()
                elif arrangement[2][1]== "Top left":
                  goto(0, 250)
                  BlockD()
                elif arrangement[2][1]== "Bottom right":
                  goto(250, 0)
```

```
                    BlockD()
                  elif arrangement[2][1] == "Top right":
                    goto(250, 250)
                    BlockD()

                elif arrangement[2][2] == "Down":
                  setheading(270)
                  if arrangement[2][1] == "Bottom left":
                    goto(-250, 250)
                    BlockD()
                  elif arrangement[2][1]== "Top left":
                    goto(-250, 500)
                    BlockD()
                  elif arrangement[2][1]== "Bottom right":
                    goto(0, 250)
                    BlockD()
                  elif arrangement[2][1] == "Top right":
                    goto(0, 500)
                    BlockD()

                elif arrangement[2][2] == "Left":
                  setheading(180)
                  if arrangement[2][1] == "Bottom left":
                    goto(0, 250)
                    BlockD()
                  elif arrangement[2][1]== "Top left":
                    goto(0, 500)
                    BlockD()
                  elif arrangement[2][1]== "Bottom right":
                    goto(250, 250)
                    BlockD()
                  elif arrangement[2][1] == "Top right":
                    goto(250, 500)
                    BlockD()

                elif arrangement[2][2] == "Right":
                  setheading(0)
                  if arrangement[2][1] == "Bottom left":
                    goto(-250, 0)
                    BlockD()
                  elif arrangement[2][1]== "Top left":
                    goto(-250, 250)
                    BlockD()
                  elif arrangement[2][1]== "Bottom right":
                    goto(0, 0)
                    BlockD()
                  elif arrangement[2][1] == "Top right":
                    goto(0, 250)
                    BlockD()
    else:
        pass

##########----------------------END OF  IF ARRANGEMENT[2] ---------------
------##########

##########------------------------- IF ARRANGEMENT[3] -------------------
------##########
######------ ARRANGEMENT[3] for BLOCK A ------######
```

```
if len(arrangement) >= 4:
    if arrangement[3][0] == "Block A":
      if arrangement[3][2] == "Up":
        setheading(90)
        if arrangement[3][1] == "Bottom left":
          goto(0, 0)
          BlockA()
        elif arrangement[3][1]== "Top left":
          goto(0, 250)
          BlockA()
        elif arrangement[3][1]== "Bottom right":
          goto(250, 0)
          BlockA()
        elif arrangement[3][1] == "Top right":
          goto(250, 250)
          BlockA()

      elif arrangement[3][2] == "Down":
        setheading(270)
        if arrangement[3][1] == "Bottom left":
          goto(-250, 250)
          BlockA()
        elif arrangement[3][1]== "Top left":
          goto(-250, 500)
          BlockA()
        elif arrangement[3][1]== "Bottom right":
          goto(0, 250)
          BlockA()
        elif arrangement[3][1] == "Top right":
          goto(0, 500)
          BlockA()

      elif arrangement[3][2] == "Left":
        setheading(180)
        if arrangement[3][1] == "Bottom left":
          goto(0, 250)
          BlockA()
        elif arrangement[3][1]== "Top left":
          goto(250, 500)
          BlockA()
        elif arrangement[3][1]== "Bottom right":
          goto(250, 250)
          BlockA()
        elif arrangement[3][1] == "Top right":
          goto(250, 500)
          BlockA()

      elif arrangement[3][2] == "Right":
        setheading(0)
        if arrangement[3][1] == "Bottom left":
          goto(-250, 0)
          BlockA()
        elif arrangement[3][1]== "Top left":
          goto(-250, 250)
          BlockA()
        elif arrangement[3][1]== "Bottom right":
          goto(0, 0)
          BlockA()
```

```
            elif arrangement[3][1] == "Top right":
              goto(0, 250)
              BlockA()


######------ ARRANGEMENT[3] for BLOCK B ------######
        elif arrangement[3][0] == "Block B":
                if arrangement[3][2] == "Up":
                  setheading(90)
                  if arrangement[3][1] == "Bottom left":
                    goto(0, 0)
                    BlockB()
                  elif arrangement[3][1]== "Top left":
                    goto(0, 250)
                    BlockB()
                  elif arrangement[3][1]== "Bottom right":
                    goto(250, 0)
                    BlockB()
                  elif arrangement[3][1] == "Top right":
                    goto(250, 250)
                    BlockB()

                elif arrangement[3][2] == "Down":
                  setheading(270)
                  if arrangement[3][1] == "Bottom left":
                    goto(-250, 250)
                    BlockB()
                  elif arrangement[3][1]== "Top left":
                    goto(-250, 500)
                    BlockB()
                  elif arrangement[3][1]== "Bottom right":
                    goto(0, 250)
                    BlockB()
                  elif arrangement[3][1] == "Top right":
                    goto(0, 500)
                    BlockB()

                elif arrangement[3][2] == "Left":
                  setheading(180)
                  if arrangement[3][1] == "Bottom left":
                    goto(0, 250)
                    BlockB()
                  elif arrangement[3][1]== "Top left":
                    goto(0, 500)
                    BlockB()
                  elif arrangement[3][1]== "Bottom right":
                    goto(250, 250)
                    BlockB()
                  elif arrangement[3][1] == "Top right":
                    goto(250, 500)
                    BlockB()

                elif arrangement[3][2] == "Right":
                  setheading(0)
                  if arrangement[3][1] == "Bottom left":
                    goto(-250, 0)
                    BlockB()
                  elif arrangement[3][1]== "Top left":
```

```
                goto(-250, 250)
                BlockB()
              elif arrangement[3][1]== "Bottom right":
                goto(0, 0)
                BlockB()
              elif arrangement[3][1] == "Top right":
                goto(0, 250)
                BlockB()


######------ ARRANGEMENT[3] for BLOCK C ------######
        elif arrangement[3][0] == "Block C":
                if arrangement[3][2] == "Up":
                  setheading(90)
                  if arrangement[3][1] == "Bottom left":
                    goto(0, 0)
                    BlockC()
                  elif arrangement[3][1]== "Top left":
                    goto(0, 250)
                    BlockC()
                  elif arrangement[3][1]== "Bottom right":
                    goto(250, 0)
                    BlockC()
                  elif arrangement[3][1] == "Top right":
                    goto(250, 250)
                    BlockC()

                elif arrangement[3][2] == "Down":
                  setheading(270)
                  if arrangement[3][1] == "Bottom left":
                    goto(-250, 250)
                    BlockC()
                  elif arrangement[3][1]== "Top left":
                    goto(-250, 500)
                    BlockC()
                  elif arrangement[3][1]== "Bottom right":
                    goto(0, 250)
                    BlockC()
                  elif arrangement[3][1] == "Top right":
                    goto(0, 500)
                    BlockC()

                elif arrangement[3][2] == "Left":
                  setheading(180)
                  if arrangement[3][1] == "Bottom left":
                    goto(0, 250)
                    BlockC()
                  elif arrangement[3][1]== "Top left":
                    goto(250, 500)
                    BlockC()
                  elif arrangement[3][1]== "Bottom right":
                    goto(250, 250)
                    BlockC()
                  elif arrangement[3][1] == "Top right":
                    goto(250, 500)
                    BlockC()

                elif arrangement[3][2] == "Right":
```

```
                setheading(0)
                if arrangement[3][1] == "Bottom left":
                  goto(-250, 0)
                  BlockC()
                elif arrangement[3][1]== "Top left":
                  goto(-250, 250)
                  BlockC()
                elif arrangement[3][1]== "Bottom right":
                  goto(0, 0)
                  BlockC()
                elif arrangement[3][1] == "Top right":
                  goto(0, 250)
                  BlockC()

######------ ARRANGEMENT[3] for BLOCK D ------######
        elif arrangement[3][0] == "Block D":
                if arrangement[3][2] == "Up":
                  setheading(90)
                  if arrangement[3][1] == "Bottom left":
                    goto(0, 0)
                    BlockD()
                  elif arrangement[3][1]== "Top left":
                    goto(0, 250)
                    BlockD()
                  elif arrangement[3][1]== "Bottom right":
                    goto(250, 0)
                    BlockD()
                  elif arrangement[3][1] == "Top right":
                    goto(250, 250)
                    BlockD()

                elif arrangement[3][2] == "Down":
                  setheading(270)
                  if arrangement[3][1] == "Bottom left":
                    goto(-250, 250)
                    BlockD()
                  elif arrangement[3][1]== "Top left":
                    goto(-250, 500)
                    BlockD()
                  elif arrangement[3][1]== "Bottom right":
                    goto(0, 250)
                    BlockD()
                  elif arrangement[3][1] == "Top right":
                    goto(0, 500)
                    BlockD()

                elif arrangement[3][2] == "Left":
                  setheading(180)
                  if arrangement[3][1] == "Bottom left":
                    goto(0, 250)
                    BlockD()
                  elif arrangement[3][1]== "Top left":
                    goto(0, 500)
                    BlockD()
                  elif arrangement[3][1]== "Bottom right":
                    goto(250, 250)
                    BlockD()
                  elif arrangement[3][1] == "Top right":
```

```
                    goto(250, 500)
                    BlockD()

                elif arrangement[3][2] == "Right":
                  setheading(0)
                  if arrangement[3][1] == "Bottom left":
                    goto(-250, 0)
                    BlockD()
                  elif arrangement[3][1]== "Top left":
                    goto(-250, 250)
                    BlockD()
                  elif arrangement[3][1]== "Bottom right":
                    goto(0, 0)
                    BlockD()
                  elif arrangement[3][1] == "Top right":
                    goto(0, 250)
                    BlockD()
    else:
        pass


##########----------------------END OF  IF ARRANGEMENT[3] ---------------
------##########


########------------End of if statements ------------########

#
#-------------------------------------------------------------------------#


#-----Main Program-------------------------------------------------------#
#
# This main program sets up the background, ready for you to start
# drawing your jigsaw pieces.  Do not change any of this code except
# where indicated by comments marked '*****'.
#

# Set up the drawing canvas
create_drawing_canvas()

# Control the drawing speed
# ***** Modify the following argument if you want to adjust
# ***** the drawing speed
speed('fastest')

# Decide whether or not to show the drawing being done step-by-step
# ***** Set the following argument to False if you don't want to wait
# ***** while the cursor moves around the screen
tracer(True)

# Give the window a title
# ***** Replace this title with one that describes the picture
# ***** produced by stacking your blocks correctly
title('The Avengers')

# Display the corner and centre coordinates of the places where
```

```
# the blocks must be placed
# ***** If you don't want to display the coordinates change the
# ***** arguments below to False
mark_coords(True, True)

### Call the student's function to display the stack of blocks
### ***** Change the argument to this function to test your
### ***** code with different data sets
stack_blocks(arrangement_99)

# Exit gracefully
release_drawing_canvas()

#
#-------------------------------------------------------------------------#
```