“””For the ﬁnal program, in addition to the drawTree function developed in the previous section,

we need an initialization function. Further, it is often worthwhile to have an entry-point

function that does initialization and then calls the function of interest. All the functions are

also carefully commented.”””

def drawTree ( segments , size ):

“””

drawTree recursively draws the tree .

segments -- NonNegInteger ;

number of line segments from the base of the tree to

the end of any branch should be integral and non - negative .

size -- PosNumber ;

length of tree " trunk " to draw should be ( strictly ) positive .

5pre - conditions : segments >= 0 , size > 0.

turtle is at base of tree ,

turtle is facing along trunk of tree ,

turtle is pen - down .

post - conditions : a segments - level tree was drawn on the canvas ,

turtle is at base of tree ,

turtle is facing along trunk of tree ,

turtle is pen - down .

"""

if segments == 0:

# base case : draw nothing

pass

else :

# recursive case : draw trunk and two sub - trees

forward ( size )

left ( 45 )

drawTree ( segments - 1 , size / 2 )

right ( 90 ) # turn 45 degrees twice

drawTree ( segments - 1 , size / 2 )

left ( 45 )

forward ( - size )

def initWorld ( size ):

"""

initWorld initializes the drawing by establishing its pre - conditions .

size -- PosNumber ;

length of tree trunk to draw should be positive .

pre - conditions :

post - conditions : coordinate system

is ( -2\*| size | , -2\*| size |) at lower - left

to (2\*| size | , 2\*| size |) at upper - right .

turtle is at origin ,

turtle is facing North ,

turtle is pen - down .

"""

margin = 2 # provides canvas boundary

setup ( 600 , 600 )

setworldcoordinates ( -2\* abs ( size ) - margin , -2\* abs ( size ) - margin , \ 2\* abs ( size ) + margin , 2\* abs ( size ) + margin )

home () # turtle is at origin , facing east , pen - down

left ( 90 ) # turtle is facing North

down () # turtle is pen - down

pensize ( 2 )

def initWorldAndDrawTree ( segments , size ):

"""

initWorldAndDrawTree prints a message , initializes the world ,

draws an instance of the recursive tree , and waits for ENTER .

segments -- NonNegInteger ;

number of line segments from the base of the tree to

6the end of any branch should be integral and non - negative .

size -- PosNumber ;

length of tree " trunk " to draw should be ( strictly ) positive .

message -- String ;

message to display

"""

print ( " Drawing recursive tree with " , ( segments , size ))

initWorld ( size )

drawTree ( segments , size )

update ()

input ( " Hit ENTER to quit ." )

bye ()

“””

Note that the drawTree expects the turtle’s position and orientation to be at the base of the

tree and facing along the trunk. This is called a pre-condition because a speciﬁc state is

assumed to have been established before the function begins its execution.

Also, note that we are careful to return the turtle to its initial position and orientation (at

the base of the tree and facing along the trunk) after drawing the tree. The turtle must

be at exactly the same position and in exactly the same orientation as at the beginning

of the drawing. Suppose otherwise: after drawing the left smaller tree, the turtle would

not be at the crotch of the tree (the base of the smaller tree) and we would draw the

right tree starting somewhere other than the crotch. (See what happens if you remove the

“move by length -size units” from the pseudocode of drawTree.) This is called a postcondition because a speciﬁc state is guaranteed to have been established after the function

ﬁnishes its execution.

We should always document pre- and post-conditions for functions. This allows other programmers to know what needs to be done before calling the function and what can be

expected after calling the function. Beware: if you call a function without satisfying its

pre-conditions, all bets are oﬀ!”””