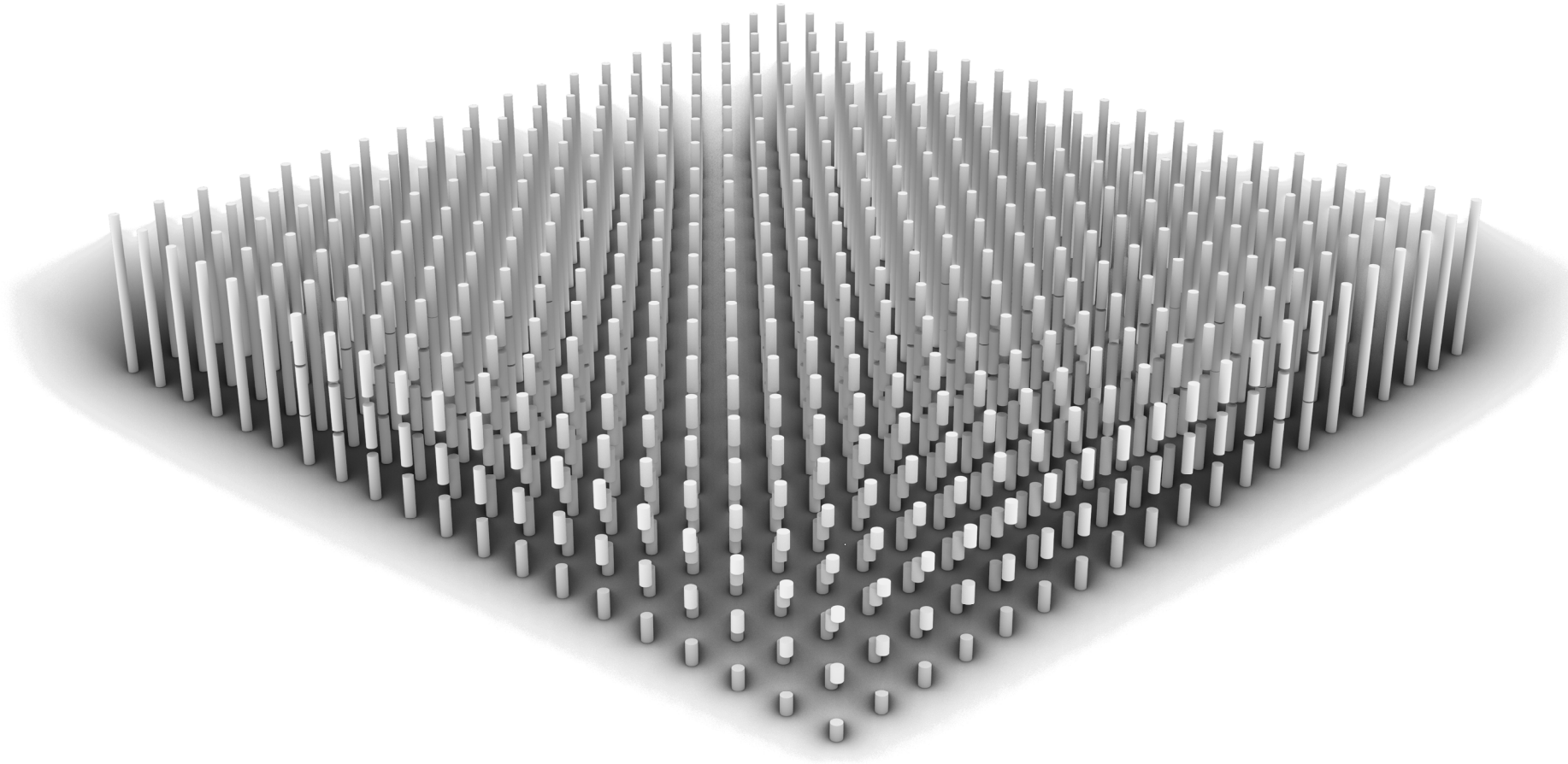

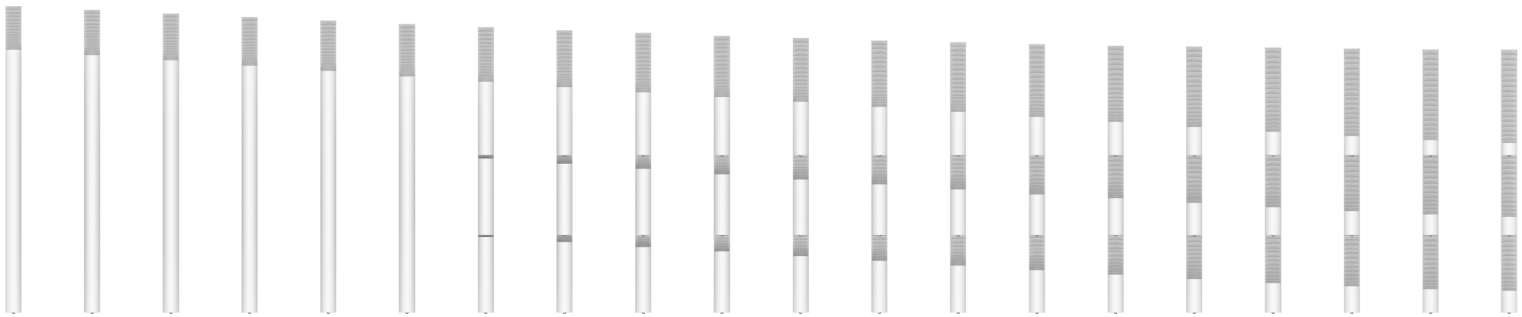

```
0   ########################################
1   ########################################
2
3   #ASSIGNMENT_2_2
4   #NAME: Maho Kobayashi
5
6   ########################################
7   ########################################
8
9   #INSTRUCTIONS:
10  ##All work should be done as 2D LINE WORK
11  ##Format as LETTER SIZE (8.5" X 11") in LANDSCAPE
12  ##Pay special attention to your LINE WIDTH
13  ##Submit as a SINGLE PDF FILE, not neccessarily 1 pg
14  ##First PNG/JPG in attachments will be cover image
15
16  ##The assignment should involve the use of . . .
17  ###Lists
18  ###Iteration
19  ###Conditional Execution
20
21  ##############################
22
23  #DELIVERABLES:
24  ##PDF
25  ##code (saved in RTF = Rich Text Format)
26  ##both(?)should be uploaded to the Gallery Site
27
28  ########################################
29  ########################################
30
31  #RHINOSCRIPT REFERENCES:
32  #https://developer.rhino3d.com/
33
34  ##############################
35
36  #PYTHON REFERENCES:
37
38  ##.append()**************.sort()**
39  ##.reverse()**************.pop()**
40  ##len()**
41
42  ##.append()
43  ###"To add an object to an already existing collections type,
44  ###for instance. This is where the append method in Python
45  ###shows its value. Append in Python is a pre-defined
46  ###method used to add a single item to certain collection types"

47  ###(https://www.simplilearn.com/ Google)
48
49  #REF: "bone structure: example 02" / 12:11 / rs.HideObject(ID)
50
51  ########################################
52  ########################################
53
54  #BRING IN LIBRARIES
55  import rhinoscriptsyntax as rs
56
```

```
56
57  ##############################
58
59  #STEP_0: CREATE EMPTY LIST
60
61  ptList = []
62
63  #STEP_1: SET THE ATTRACTOR POINT
64
65  attrPt = rs.GetObject('select an attractor point',rs.filter.point)
66
67  #STEP_2: CREATE 2D POINT MATRIX
68  for i in range(20):
69          for j in range(20):
70                  for k in range(3):
71
72                          #KEY
73                          ##define x in terms of i
74                          ##define y in terms of j
75                          x = i
76                          y = j
77                          z = k
78
79                          rs.AddPoint(x,y,z)
80                          #NOTES ABOUT 'rs.AddPoint()'
81                          ##rs.AddPoint(stop)
82                          ##rs.AddPoint(start,stop)
83                          ##rs.AddPoint(start,stop,step)
84
85                          #rs.AddTextDot((x,y,z),(x,y,z))
86                          #NOTES ABOUT 'rs.AddTextDot()
87                          ##if do not need the z, exclude z
88
89                          ptList.append((x,y,z))
90                          #print(ptList_0)
91
92  for i in range(len(ptList)):
93  #       print i, ':', ptList[i]
94          #rs.AddTextDot(i, ptList[i])#<--- unhashtag to see labeled as
      index #
95  ##NOTES: loop through point list and print out index number and
    values
96
97          ##############################
98          #STEP 3: CREATING TRANSFORMATION OF GEOMETRY
99          #measure distance between attracor point and current point
    in the list
100         distance = rs.Distance(ptList[i], attrPt)
101         print distance/20
102         #create circle using distance value as radius
103 #        rs.AddCylinder(ptList[i], distance/80,1/8)
104 #        rs.AddSphere(ptList[i], distance/180)
105 #        rs.AddCircle(ptList[i], distance/2)
106         rs.AddCylinder(ptList[i], distance/15,1/10)
107
108
```