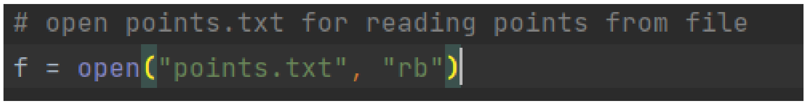
Nathalie Agustin

nagusti1

JHU EP 606.206 - Introduction to Programming Using Python

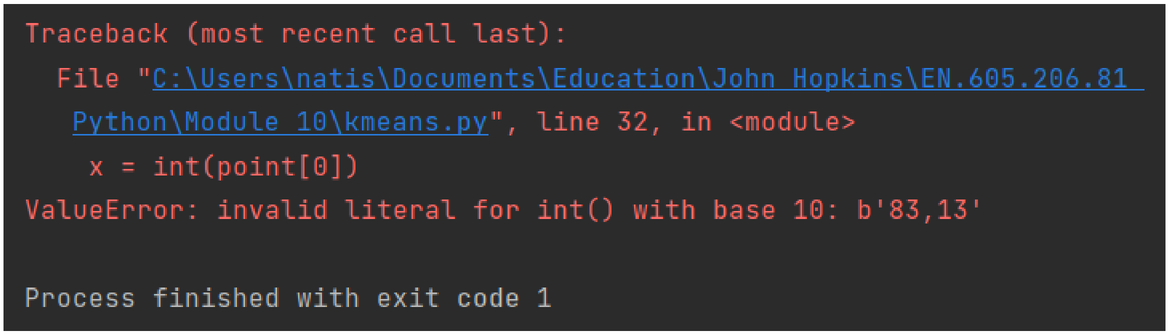
Module 10: Debugging K-means Clustering

Line 9 of unpatched kmeans.py (1)



```
# open points.txt for reading points from file
f = open("points.txt", "rb")
```

1. Bug Description:
    a. This is a bug because the data from the input file is being read in as binary.
    b. It is triggered when the file is open.
    c. When running the program, the values for an integer of base ten come out as binary. See screenshot below.
    d. ValueError inherits from Exception, and it is raised on runtime.
2. Type of Error: Logical
3. Bug Fix:
    a. Before: `f = open(“points.txt”, “rb”)`
    b. After: `f = open(“points.txt”, r”)`
4. Fix Description: Opening in “r” (read) mode opens the file as encoded strings rather than in binary mode.
5.

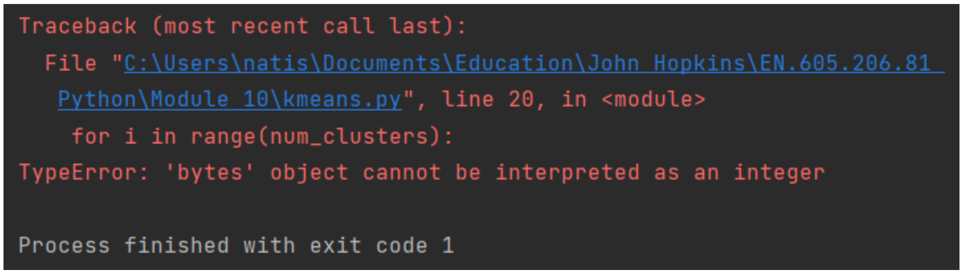


```
Traceback (most recent call last):
  File "C:\Users\natis\Documents\Education\John Hopkins\EN.605.206.81 
   Python\Module 10\kmeans.py", line 32, in <module>
    x = int(point[0])
ValueError: invalid literal for int() with base 10: b'83,13'

Process finished with exit code 1
```

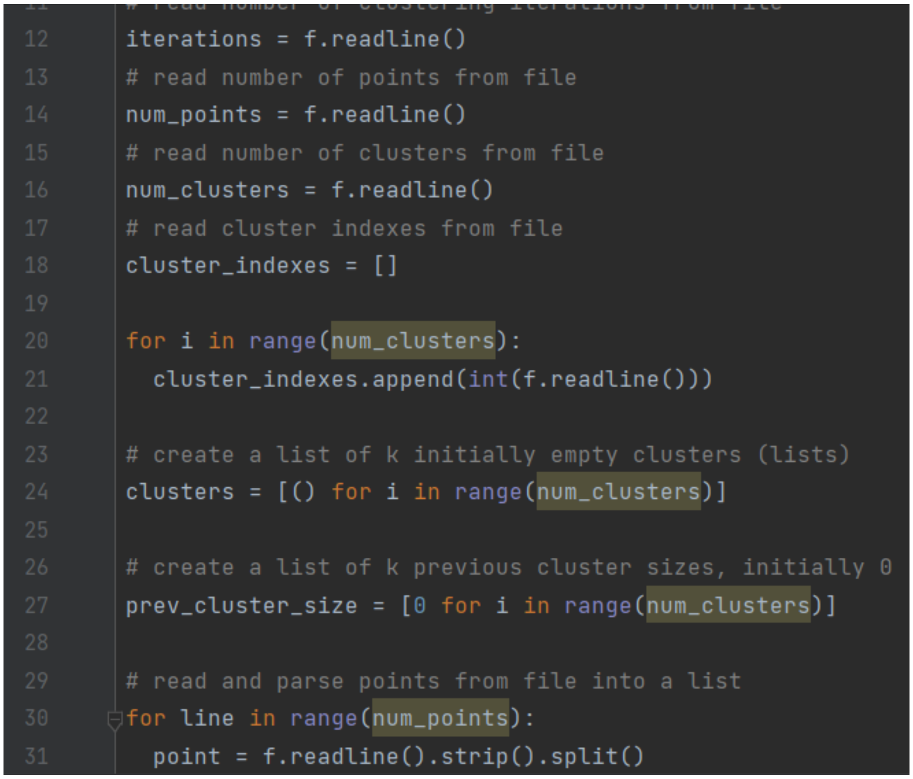
Lines 20, 24, 27, and 30 of unpatched kmeans.py. (4)

1. Bug Description:
    a. It is a bug because we want to assess the data from the file as integers rather than a string of bytes, since readline() returns a string by default.
    b. It is triggered on run-time, where the for loops in the code will iterate across the string characters of text from the input file rather than evaluate the data as numbers.

   c. I found it through the PyCharm IDE which automatically highlights errors.
   d. Exception caused?
2. Type of Error: Logical
3. Bug Fix:
   a. Before:
      `f.readline()`
   b. After:
      `int(f.readline())`
4. Fix Description
   a. Type cast iterations, num_points, and num_clusters as integers rather than leaving them as a string of bytes.
5. Screenshot of exception caused by the bug/error



```
Traceback (most recent call last):
  File "C:\Users\natis\Documents\Education\John Hopkins\EN.605.206.81 
   Python\Module 10\kmeans.py", line 20, in <module>
    for i in range(num_clusters):
TypeError: 'bytes' object cannot be interpreted as an integer

Process finished with exit code 1
```



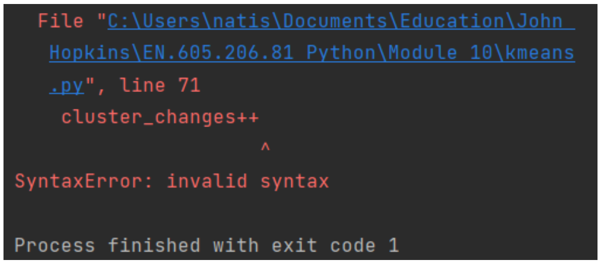
```
11        # read number of clustering iterations from file
12        iterations = f.readline()
13        # read number of points from file
14        num_points = f.readline()
15        # read number of clusters from file
16        num_clusters = f.readline()
17        # read cluster indexes from file
18        cluster_indexes = []
19
20        for i in range(num_clusters):
21          cluster_indexes.append(int(f.readline()))
22
23        # create a list of k initially empty clusters (lists)
24        clusters = [() for i in range(num_clusters)]
25
26        # create a list of k previous cluster sizes, initially 0
27        prev_cluster_size = [0 for i in range(num_clusters)]
28
29        # read and parse points from file into a list
30        for line in range(num_points):
31          point = f.readline().strip().split()
```

Line 32 (1)

1. Bug Description
    a. It is a bug because the points input file is not being parsed as we want it to be. The comma still remains even though the extra whitespace has been stripped.
    b. It is triggered when it is passed through the Python interpreter.
    c. I found it by attempting to run the kmeans.py program on the PyCharm IDE.
    d. It did not raise an exception.
2. Type of Error: Logical
3. Bug Fix:
    a. Before:
    ```
    point = f.readline().strip().split()
    ```
    b. After:
    ```
    point = f.readline().strip().split(',')
    ```
4. Fix Description:
   A comma is passed to the split function to specify where the values will be separated. This allows for the coordinate values to be used independently.
5. Screenshot: N/A

Line 71 (1)

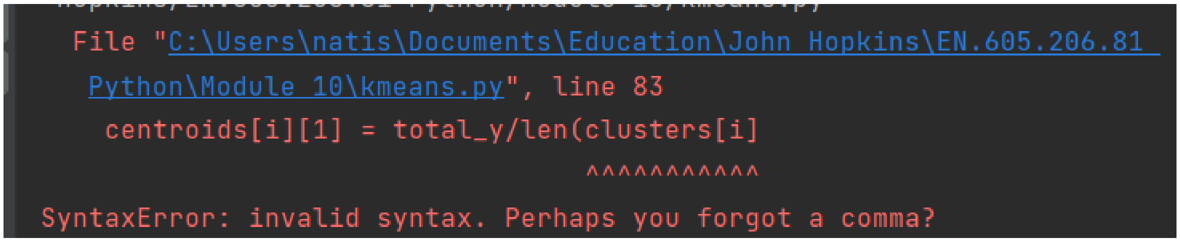
6. Bug Description
    a. It is a bug because Python does not allow incrementing by one using the ++ operator. The ++ operator does not exist, and the lexical analyzer cannot parse it as a token.
    b. It is triggered when it is passed through the Python interpreter.
    c. I found it by attempting to run the kmeans.py program.
    d. SyntaxError raised.
7. Type of Error: Lexical
8. Bug Fix:
    a. Before:
    ```
    cluster_changes ++
    ```
    b. After:
    ```
    cluster_changes += 1
    ```
9. Fix Description:
   The += operator is a valid way to increment the value of cluster_changes by 1 without creating a lexical error.
10. Screenshot:



```
  File "C:\Users\natis\Documents\Education\John 
  Hopkins\EN.605.206.81 Python\Module 10\kmeans
  .py", line 71
    cluster_changes++
                     ^
SyntaxError: invalid syntax

Process finished with exit code 1
```

Line 83 (1)

1. Bug Description
   a. It is a bug because an expected end parenthesis should be in this line.
   b. It is triggered when it is passed through the Python interpreter.
   c. I found it by attempting to run the kmeans.py program.
   d. SyntaxError raised.
2. Type of Error: Syntax
3. Bug Fix:
   a. Before:
      `centroids[i][1] = total_y/len(clusters[i]`
   b. After:
      `centroids[i][1] = total_y/len(clusters[i])`
4. Fix Description:
   Adding a parentheses closes the expected syntax for the len() function.
5. Screenshot:



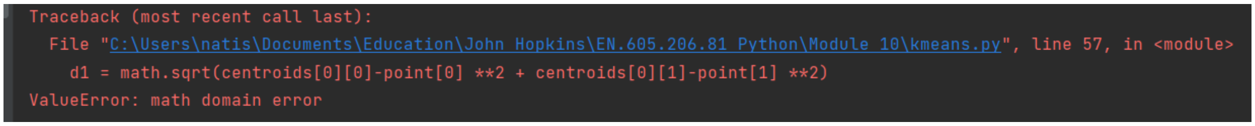
```
  File "C:\Users\natis\Documents\Education\John Hopkins\EN.605.206.81 
  Python\Module 10\kmeans.py", line 83
    centroids[i][1] = total_y/len(clusters[i]
                                  ^^^^^^^^^^^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

Line 53-56 (4):

1. Bug Description
   a. It is a bug because the order of operations is not followed as intended for calculating distance.
   b. It is triggered when attempting to evaluate the distance between the centroids and the points in scrutiny.
   c. It was found by running the code in the PyCharm IDE.

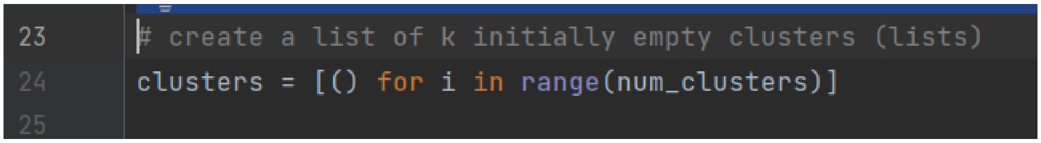
   d. An ValueError: math domain error is raised.

2. Type of Error: Semantic
3. Bug Fix:
   a. Before: `math.sqrt(centroids[0][0]-point[0]**2 + centroids[0][1]-point[1]**2)`
   b. After: `math.sqrt((centroids[0][0]-point[0]**2) + (centroids[0][1]-point[1]**2))`
4. Fix Description: Parenthesis are added around the centroid point calculations in order to have those operations executed first before having them added together.
5. Screenshot:



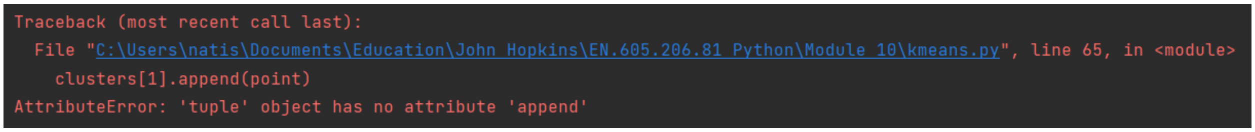
```
Traceback (most recent call last):
  File "C:\Users\natis\Documents\Education\John Hopkins\EN.605.206.81 Python\Module 10\kmeans.py", line 57, in <module>
    d1 = math.sqrt(centroids[0][0]-point[0] **2 + centroids[0][1]-point[1] **2)
ValueError: math domain error
```

Line 24 (1)



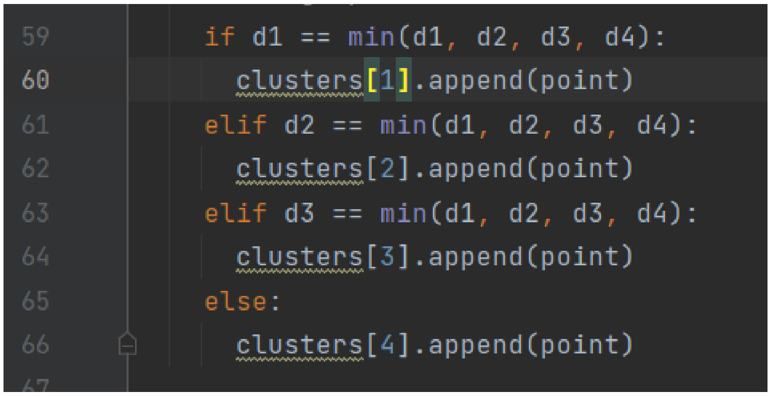
```
23  # create a list of k initially empty clusters (lists)
24  clusters = [() for i in range(num_clusters)]
25
```

1. Bug Description
   a. It is a bug because a list of tuples is created rather than a list of lists.
   b. It is triggered when the interpreter is attempting to perform actions that are capable on a list but are not possible with a tuple.
   c. It was found by attempting to run the code.
   d. An AttributeError is raised.
2. Type of Error: Logical
3. Bug Fix:
   a. Before: clusters = `[() for i in range(num_clusters)]`
   b. After: `[[] for i in range(num_clusters)]`
4. Fix Description: By adding two closed brackets rather than two closed parentheses, the clusters list is defined as a list of lists rather than a list of tuples.
5. Screenshot:



```
Traceback (most recent call last):
  File "C:\Users\natis\Documents\Education\John Hopkins\EN.605.206.81 Python\Module 10\kmeans.py", line 65, in <module>
    clusters[1].append(point)
AttributeError: 'tuple' object has no attribute 'append'
```

Line 66 (4)



```
59        if d1 == min(d1, d2, d3, d4):
60          clusters[1].append(point)
61        elif d2 == min(d1, d2, d3, d4):
62          clusters[2].append(point)
63        elif d3 == min(d1, d2, d3, d4):
64          clusters[3].append(point)
65        else:
66          clusters[4].append(point)
```

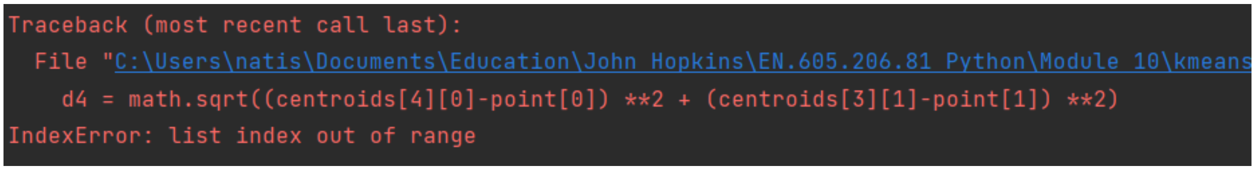
1. Bug Description
    a. It is a bug because in the given points file, there are only 4 clusters. So the clusters list should have a size of 4, but the indexes should start from (0-3) rather than (1-4)
    b. It is triggered when the interpreter is attempting to add items to the clusters list at index 4 when the last index of the clusters list ends at 3.
    c. It was found by attempting to run the code.
    d. An IndexError is raised.
2. Type of Error: Logical
3. Bug Fix:
    a. Before:

```
if d1 == min(d1, d2, d3, d4):
  clusters[1].append(point)
elif d2 == min(d1, d2, d3, d4):
  clusters[2].append(point)
elif d3 == min(d1, d2, d3, d4):
  clusters[3].append(point)
else:
  clusters[4].append(point)
```

    b. After:

```
if d1 == min(d1, d2, d3, d4):
  clusters[0].append(point)
elif d2 == min(d1, d2, d3, d4):
  clusters[1].append(point)
elif d3 == min(d1, d2, d3, d4):
  clusters[2].append(point)
else:
  clusters[3].append(point)
```

4. Fix Description: Changing the indexes of the clusters being populated with points from 1-4 to 0-3 prevents an index being accessed that is out of bounds.
5. Screenshot:



```
Traceback (most recent call last):
  File "C:\Users\natis\Documents\Education\John Hopkins\EN.605.206.81 Python\Module 10\kmeans
    d4 = math.sqrt((centroids[4][0]-point[0]) **2 + (centroids[3][1]-point[1]) **2)
IndexError: list index out of range
```

Line 78 and 79 (2):

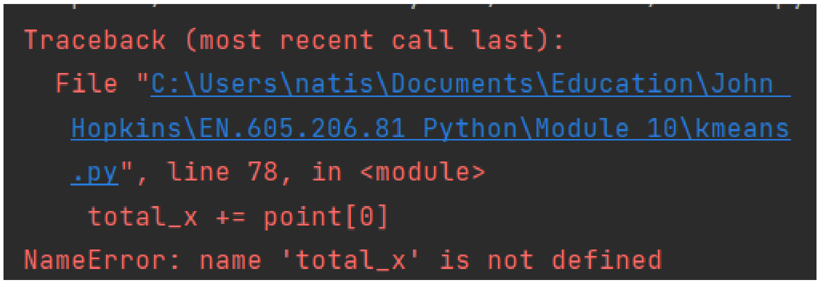
1. Bug Description:
    a. It is a bug because the code is attempting to modify a value that does not exist yet.
    b. It is triggered when the code is attempting to add point values to total_x and total_y.
    c. It was found by running the code on the PyCharm IDE.
    d. A NameError is raised.
2. Type of Error: Semantic
3. Bug Fix:
    a. Before:

```
  for i in range(len(clusters)):
    # sum the x and y coordinates of each point in the
current cluster
    for point in clusters[i]:
      total_x += point[0]
      total_y += point[1]
```

    b. After

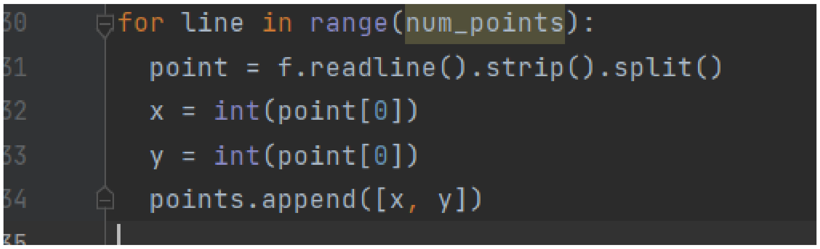
```
  for i in range(len(clusters)):
    total_x = 0
    total_y = 0

    # sum the x and y coordinates of each point in the
current cluster
    for point in clusters[i]:
      total_x += point[0]
      total_y += point[1]
```

4. Fix Description
   Before the for loop that adds all the x and y values, total_x and total_y is defined starting with a value of zero.
5. Screenshot



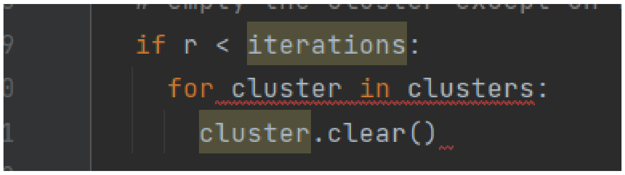
```
Traceback (most recent call last):
  File "C:\Users\natis\Documents\Education\John 
   Hopkins\EN.605.206.81 Python\Module 10\kmeans
   .py", line 78, in <module>
    total_x += point[0]
NameError: name 'total_x' is not defined
```

Line 33:



```
30      for line in range(num_points):
31          point = f.readline().strip().split()
32          x = int(point[0])
33          y = int(point[0])
34          points.append([x, y])
```

1. Bug Description:
    a. It is a bug because the value of “y” is being assigned the values for “x”.
    b. It is triggered when running the code to assign the values x and y after reading the points from the file.
    c. It was found when printing the result of the code and observing that the x and y values are identical.
    d. No Exceptions are raised.
2. Type of Error: Logical
3. Bug Fix:
    a. Before: `y = int(point[0])`
    b. After: `y = int(point[1])`
4. Fix Description:
   Rather than assigning y the first value in the coordinate pairs, y is assigned the second value to correspond with the expected y values from the points file.
5. Screenshot: N/A

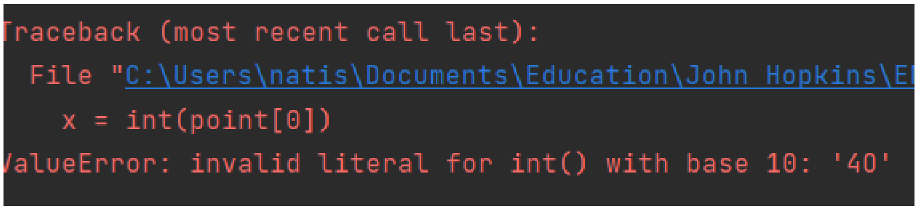


```
# empty the cluster except on
if r < iterations:
    for cluster in clusters:
        cluster.clear()
```

1. Bug Description:
    a. It is a bug because it will clear all clusters even if it is in the final iteration.
    b. It is triggered when the code is preparing the clear the lists for the next iteration of clustering.
    c. It was found when printing the result of the code and observing that there were no values in the cluster lists.
    d. No Exceptions are raised.
2. Type of Error: Logical
3. Bug Fix:
    a. Before: `if r < iterations:`
    b. After: `if r < iterations - 1:`
4. Fix Description:
   Rather than clearing at the last iteration, you must hold all the values from the second to last iteration. Once you establish that no change was found from the previous iteration, the previous iterations values are used as the final clusters. So we have to change r < iterations to iterations - 1.
5. Screenshot: N/A

Input Errors (2)

There's an "O" instead of a zero in "4O"



```
Traceback (most recent call last):
  File "C:\Users\natis\Documents\Education\John Hopkins\E
    x = int(point[0])
ValueError: invalid literal for int() with base 10: '4O'
```

There's a value "618" that is not separated in to x,y format.

1. Description
    a. It is a bug because the two values are not in the expected format of a point ("x,y").
    b. The first error is triggered when the interpreter attempts to type cast the "4O" as an integer. The second error is triggered when trying to split 618.
    c. I found it by attempting to run the kmeans.py program.
    d. TypeError: a bytes-like object is required, not 'str'
2. Type of Error: Input

3. Bug Fix:
    a. Before:

    ```
    x = int(point[0])
    y = int(point[0])
    points.append([x, y])
    ```

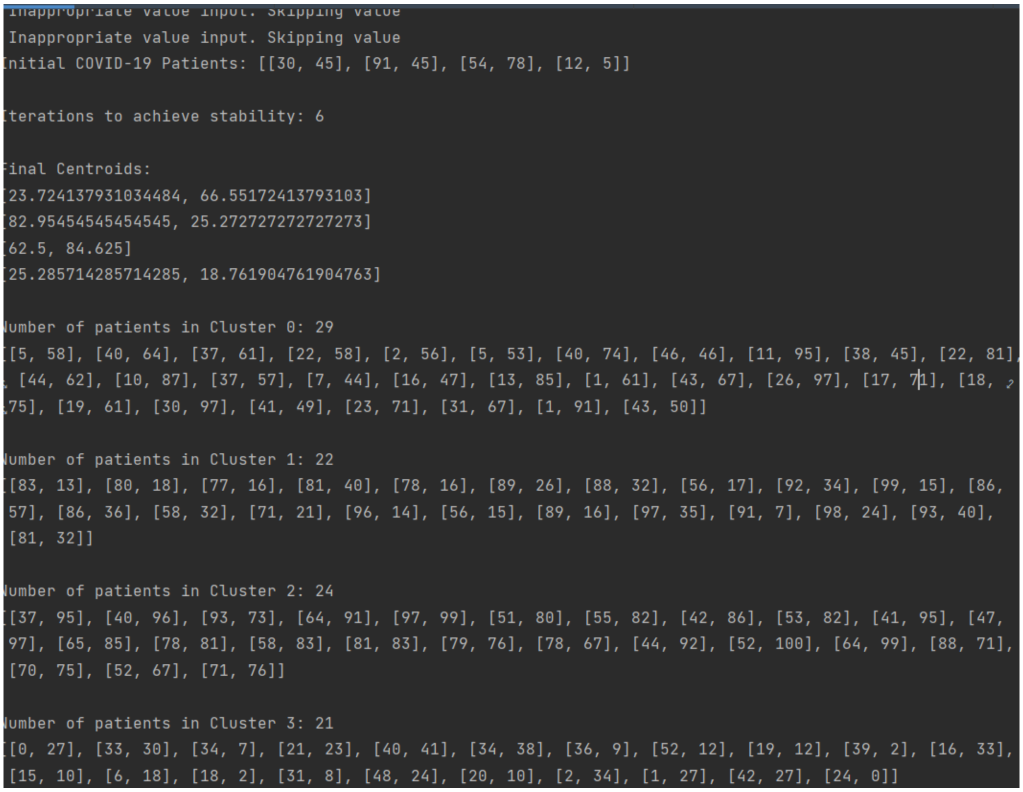
    b. After:

    ```
    try:
      x = int(point[0])
      y = int(point[1])
      points.append([x, y])
    except:
      print(f" Inappropriate value input. Skipping value")
    ```

4. Description:

Screenshot: N/A

Output Screenshot:



```
 Inappropriate value input. Skipping value
 Inappropriate value input. Skipping value
Initial COVID-19 Patients: [[30, 45], [91, 45], [54, 78], [12, 5]]

Iterations to achieve stability: 6

Final Centroids:
[23.724137931034484, 66.55172413793103]
[82.95454545454545, 25.272727272727273]
[62.5, 84.625]
[25.285714285714285, 18.761904761904763]

Number of patients in Cluster 0: 29
[[5, 58], [40, 64], [37, 61], [22, 58], [2, 56], [5, 53], [40, 74], [46, 46], [11, 95], [38, 45], [22, 81],
 [44, 62], [10, 87], [37, 57], [7, 44], [16, 47], [13, 85], [1, 61], [43, 67], [26, 97], [17, 71], [18,
75], [19, 61], [30, 97], [41, 49], [23, 71], [31, 67], [1, 91], [43, 50]]

Number of patients in Cluster 1: 22
[[83, 13], [80, 18], [77, 16], [81, 40], [78, 16], [89, 26], [88, 32], [56, 17], [92, 34], [99, 15], [86,
 57], [86, 36], [58, 32], [71, 21], [96, 14], [56, 15], [89, 16], [97, 35], [91, 7], [98, 24], [93, 40],
 [81, 32]]

Number of patients in Cluster 2: 24
[[37, 95], [40, 96], [93, 73], [64, 91], [97, 99], [51, 80], [55, 82], [42, 86], [53, 82], [41, 95], [47,
 97], [65, 85], [78, 81], [58, 83], [81, 83], [79, 76], [78, 67], [44, 92], [52, 100], [64, 99], [88, 71],
 [70, 75], [52, 67], [71, 76]]

Number of patients in Cluster 3: 21
[[0, 27], [33, 30], [34, 7], [21, 23], [40, 41], [34, 38], [36, 9], [52, 12], [19, 12], [39, 2], [16, 33],
 [15, 10], [6, 18], [18, 2], [31, 8], [48, 24], [20, 10], [2, 34], [1, 27], [42, 27], [24, 0]]
```