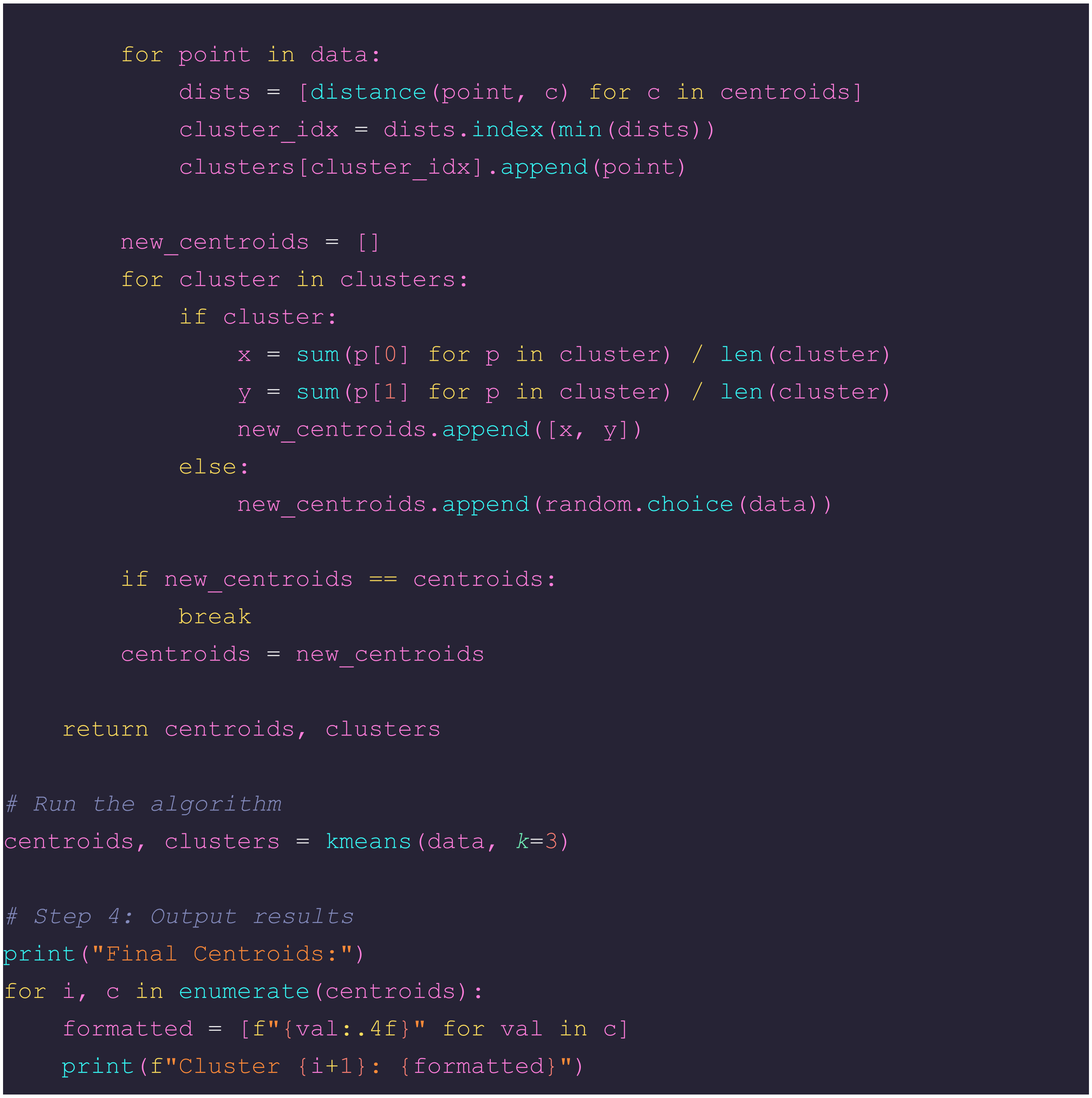
LAB TASK 10

K MEANS Clustering

Dataset used - Link

Code -

```
import csv

data = []
with open("Mall_Customers.csv", "r") as file:
    reader = csv.reader(file)
    next(reader)
    for row in reader:
        income = float(row[3])
        score = float(row[4])
        data.append([income, score])

# Check if dataset loaded properly
#print("First 5 rows of the dataset:")
#for row in data[:5]:
#    print(row)
#
#print(f"\nTotal rows loaded: {len(data)}")

import random
import math

def distance(p1, p2):
    return math.sqrt((p1[0] - p2[0]) ** 2 + (p1[1] - p2[1]) ** 2)

def kmeans(data, k=3, max_iter=100):
    centroids = random.sample(data, k)

    for _ in range(max_iter):
        clusters = [[] for _ in range(k)]
```



```
        for point in data:
            dists = [distance(point, c) for c in centroids]
            cluster_idx = dists.index(min(dists))
            clusters[cluster_idx].append(point)

        new_centroids = []
        for cluster in clusters:
            if cluster:
                x = sum(p[0] for p in cluster) / len(cluster)
                y = sum(p[1] for p in cluster) / len(cluster)
                new_centroids.append([x, y])
            else:
                new_centroids.append(random.choice(data))

        if new_centroids == centroids:
            break
        centroids = new_centroids

    return centroids, clusters

# Run the algorithm
centroids, clusters = kmeans(data, k=3)

# Step 4: Output results
print("Final Centroids:")
for i, c in enumerate(centroids):
    formatted = [f"{val:.4f}" for val in c]
    print(f"Cluster {i+1}: {formatted}")
```

Output -

```
Final Centroids:
Cluster 1: ['91.1250', '71.0000']
Cluster 2: ['25.7273', '20.2273']
Cluster 3: ['55.1692', '47.5923']
```