PROBLEM:

IMPLEMENT HIERARCHICAL CLUSTERING ALGORITHM AND DISPLAY THE CLUSTERING RESULTS USING DENDOGRAM OR VENN DIAGRAM.

THEORY:

Clustering:

Clustering is a technique in unsupervised learning where similar data points are grouped together based on certain characteristics or features. The goal is to find natural structures or patterns within the data. Clustering algorithms, such as K-Means and DBSCAN, aim to divide a dataset into groups, or clusters, where intra-cluster similarity is high and inter-cluster similarity is low. It is used in various applications such as pattern recognition, image segmentation, and customer segmentation.

Hierarchical clustering:

Hierarchical clustering is a method of clustering that organizes data into a tree-like structure. The resulting hierarchy is visualized using a **dendrogram**, which provides a clear picture of the cluster formation process. Hierarchical clustering can be categorized into:

1. Agglomerative Clustering (Bottom-Up):
    - Starts with each data point as an individual cluster.
    - Iteratively merges the closest clusters until all points are grouped into one cluster.
2. Divisive Clustering (Top-Down):
    - Starts with all data points in a single cluster.
    - Splits clusters iteratively until each point forms its own cluster.

Dendrogram

A **dendrogram** is a tree-like diagram that represents the hierarchical structure of clusters formed during hierarchical clustering. It visually depicts the process of merging or splitting clusters at various levels of similarity or dissimilarity.Key Components of a Dendrogram

1. **Leaf Nodes (Base)**: Represent individual data points or clusters before merging.
2. **Branches**: Show how clusters are merged at different levels of hierarchy.
3. **Horizontal Lines**: Indicate a cluster merge, with the height of the line representing the distance (dissimilarity) between the merged clusters.
4. **Height (y-axis)**: Represents the dissimilarity (e.g., Euclidean distance) between clusters when they were merged.

Venn diagram :

A Venn diagram is a graphical representation of the relationships between different sets or groups. It is used to illustrate how sets intersect, or overlap, and how they are distinct from each other. The diagram typically consists of overlapping circles, where

- Each circle represents a set.
- The overlapping area between circles represents the common elements of those sets.
- The non-overlapping parts of the circles represent elements unique to that set.

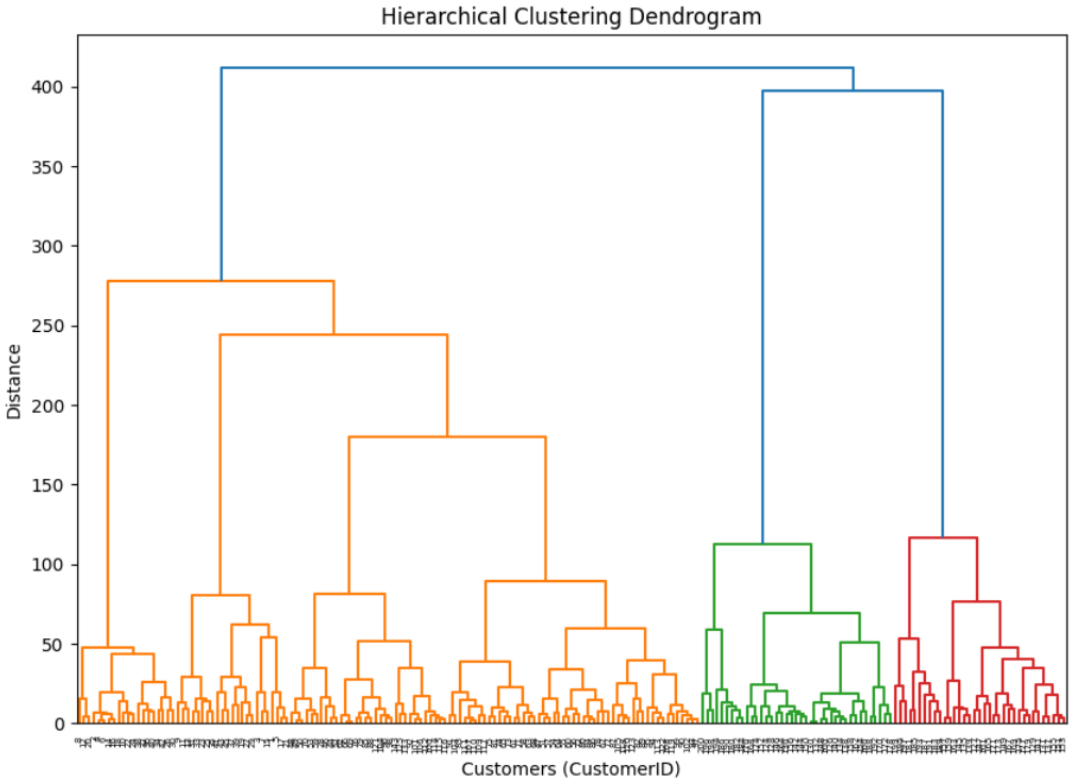
PYTHON CODE:

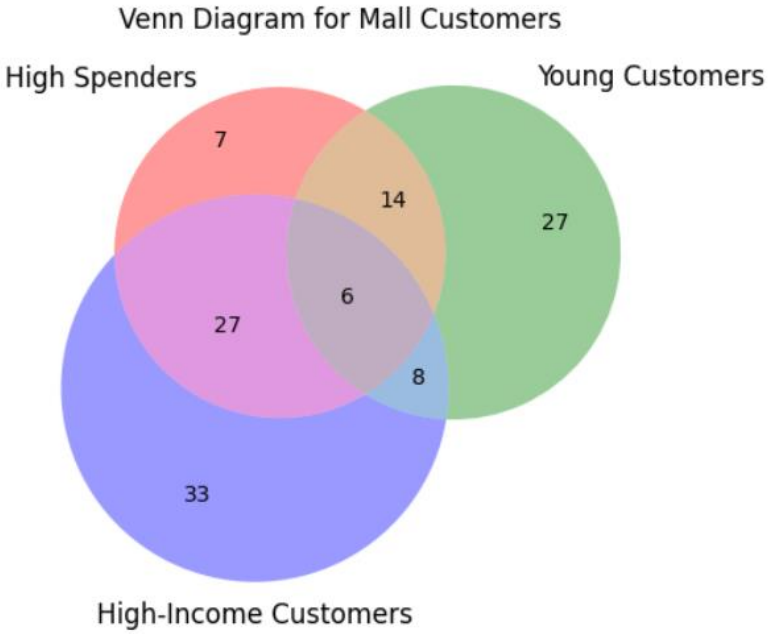
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib_venn import venn3

# Load the Mall_Customers.csv dataset
# Assuming the CSV file is in the same directory as the script
df = pd.read_csv("Mall_Customers.csv")

# Display first few rows of the dataset to understand its structure
print(df.head())

# Extract relevant columns for clustering: Age, Annual Income, Spending Score
data = df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']].values

# Perform hierarchical clustering using 'ward' linkage method
linked = linkage(data, method='ward')

# Create the dendrogram
plt.figure(figsize=(10, 7))
dendrogram(linked, labels=df['CustomerID'].astype(str).values)
plt.title("Hierarchical Clustering Dendrogram")
plt.xlabel("Customers (CustomerID)")
plt.ylabel("Distance")
plt.show()

# Venn Diagram based on specific conditions:
# Let's assume three sets of customers:
# Set A: High spenders (Spending Score > 70)
# Set B: Young customers (Age < 30)
# Set C: High-income customers (Annual Income > 70)

# Creating the sets
high_spenders = set(df[df['Spending Score (1-100)'] > 70].index)
young_customers = set(df[df['Age'] < 30].index)
high_income_customers = set(df[df['Annual Income (k$)'] > 70].index)

# Create a Venn diagram
venn3([high_spenders, young_customers, high_income_customers],
    set_labels=('High Spenders', 'Young Customers', 'High-Income Customers'))
plt.title("Venn Diagram for Mall Customers")
plt.show()
```

OUTPUT:

First few rows of the dataset:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40





EXPLANATION:

1. Importing Required Libraries

- **pandas**: To load and handle the dataset.
- **numpy**: For numerical computations.
- **matplotlib.pyplot**: To plot the dendrogram.
- **scipy.cluster.hierarchy**: To perform hierarchical clustering and create the dendrogram.
- **sklearn.preprocessing.StandardScaler**: To standardize the dataset, making it suitable for clustering.

2. Loading the Dataset

We loaded the dataset (Mall_Customers.csv) into a pandas DataFrame using the read_csv method. Then, we displayed the first few rows of the dataset to understand its structure.

3. SelectingRelevant Numerical Features

- We identified two columns, **Annual Income (k$)** and **Spending Score (1-100)**, as relevant features for clustering.
- A check was added to ensure these columns exist in the dataset. If they were missing, the program would exit with an error message.

4. Standardizing the Data

- Standardization was applied using StandardScaler to transform the data:
  - The mean of each feature was set to 0.
  - The standard deviation of each feature was set to 1.
- Standardization ensured that all features contributed equally to the clustering process, preventing bias from features with larger scales.

5. Performing Hierarchical Clustering

- We used the **Ward's method** of hierarchical clustering, which minimizes the variance within clusters at each step.
- The linkage function returned a linkage matrix, which stored information about the sequence of cluster merges and the distances at which they occurred.

6. Visualizing the Clustering with a Dendrogram

- We plotted a **dendrogram** to visualize the clustering process.
- Key improvements for clarity:
  - **Figure size**: Enlarged to 15x10 for better visibility.
  - **Leaf rotation and font size**: Rotated labels and adjusted font size to make data points readable.
  - **Axes labels**: Added descriptive labels to show data points and distances.
- The **y-axis** represented the distances (dissimilarity) at which clusters were merged.
- The **x-axis** represented individual data points.

ASSIGNMENT:

08