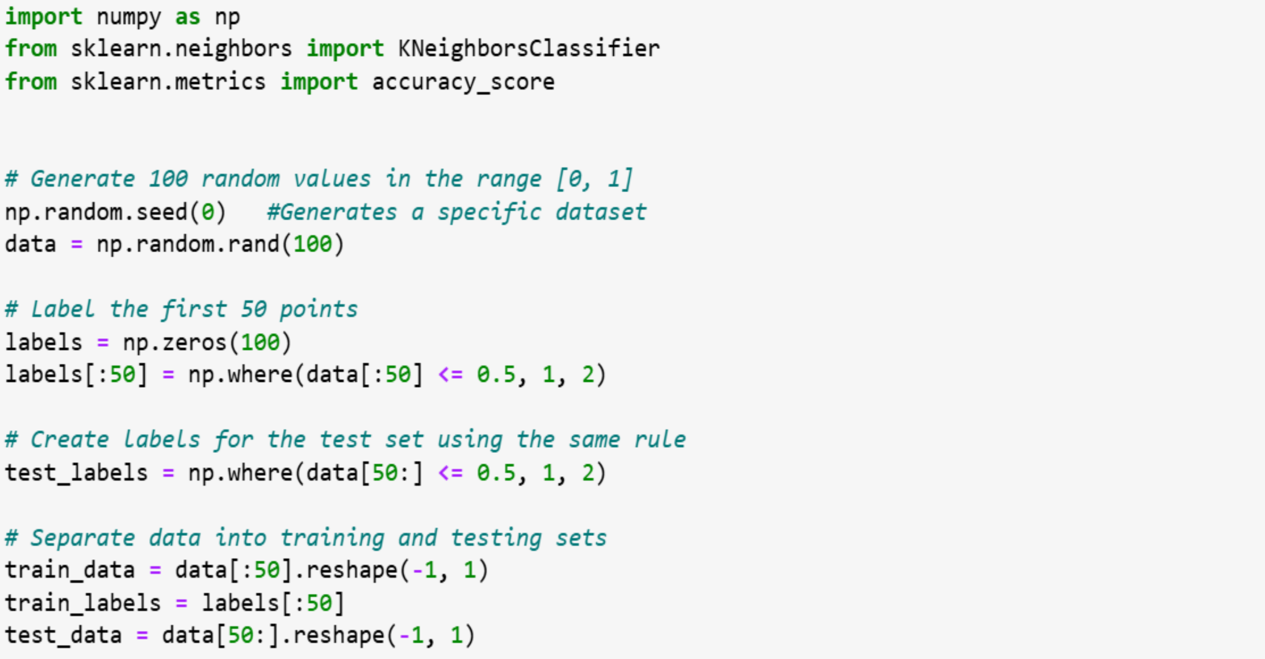
Program 5

Develop a program to implement k-Nearest Neighbour algorithm to classify the randomly generated 100 values of *x* in the range of [0,1]. Perform the following based on dataset generated.

1. Label the first 50 points {*x1,……,x50*} as follows: if (xi ≤ 0.5), then *xi* ε Class1, else *xi* ε Class2

2. Classify the remaining points, *x51,……,x100* using KNN. Perform this for *k=1,2,3,4,5,20,30*



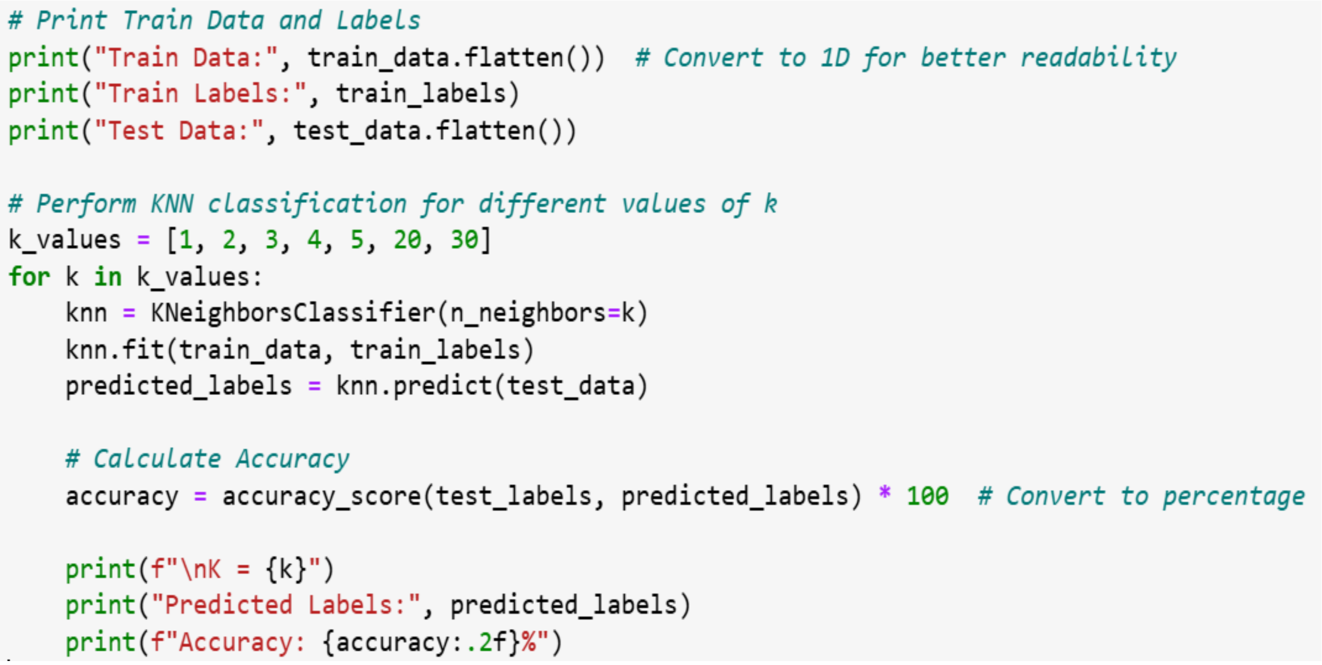
```
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score


# Generate 100 random values in the range [0, 1]
np.random.seed(0)   #Generates a specific dataset
data = np.random.rand(100)

# Label the first 50 points
labels = np.zeros(100)
labels[:50] = np.where(data[:50] <= 0.5, 1, 2)

# Create labels for the test set using the same rule
test_labels = np.where(data[50:] <= 0.5, 1, 2)

# Separate data into training and testing sets
train_data = data[:50].reshape(-1, 1)
train_labels = labels[:50]
test_data = data[50:].reshape(-1, 1)
```



```
# Print Train Data and Labels
print("Train Data:", train_data.flatten())  # Convert to 1D for better readability
print("Train Labels:", train_labels)
print("Test Data:", test_data.flatten())

# Perform KNN classification for different values of k
k_values = [1, 2, 3, 4, 5, 20, 30]
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(train_data, train_labels)
    predicted_labels = knn.predict(test_data)

    # Calculate Accuracy
    accuracy = accuracy_score(test_labels, predicted_labels) * 100  # Convert to percentage

    print(f"\nK = {k}")
    print("Predicted Labels:", predicted_labels)
    print(f"Accuracy: {accuracy:.2f}%")
```