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.

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

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

```
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter

# Step 1: Generate 100 random values in the range [0,1]
x_values = np.random.rand(100)

# Step 2: Label the first 50 points
labels = np.array(["Class1" if x <= 0.5 else "Class2" for x in x_values[:50]])

print(x_values)
print("--------------------------------------")
print(labels)

# Step 3: Define the KNN function
def knn_classify(x_train, y_train, x_test, k):
    predictions = []

    for x in x_test:
        # Compute distances from x to all x_train points
        distances = np.abs(x_train - x)

        # Get indices of k nearest neighbors
        k_nearest_indices = np.argsort(distances)[:k]

        # Get the labels of k nearest neighbors
        k_nearest_labels = y_train[k_nearest_indices]

        # Determine the most common class among neighbors
        most_common = Counter(k_nearest_labels).most_common(1)[0][0]

        # Store the predicted class
        predictions.append(most_common)
    return np.array(predictions)

# Step 4: Classify the remaining 50 points using KNN for different values of k
k_values = [1, 2, 3, 4, 5, 20, 30]
results = {}

for k in k_values:
    predicted_labels = knn_classify(x_values[:50], labels, x_values[50:], k)
    results[k] = predicted_labels

# Step 5: Visualization with clusters
plt.figure(figsize=(10, 6))
for k in k_values:
    plt.figure(figsize=(10, 6))

    # Plot labeled data
    plt.scatter(x_values[:50], [1]*50, c=["blue" if lbl == "Class1" else "red" for lbl in labels], label="Labeled Data")

    # Plot classified data
    plt.scatter(x_values[50:], [2]*50, c=["blue" if lbl == "Class1" else "red" for lbl in results[k]], label=f"Classified Data (k={k})")

    plt.xlabel("x values")
    plt.ylabel("Classified/Unclassified")
    plt.title(f"KNN Classification Clusters (k={k})")
    plt.legend()
    plt.show()

# Step 6: Print classification results
for k, preds in results.items():
    print(f"Results for k={k}:")
    print(preds)
    print("-")
```