**6. Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.**

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy\_score

from sklearn import datasets

# Load the Iris dataset

iris = datasets.load\_iris()

X = iris.data

y = iris.target

# Split the dataset into 70% training and 30% testing

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Create the KNN classifier with k=5 (you can change k as needed)

knn = KNeighborsClassifier(n\_neighbors=5)

knn.fit(X\_train, y\_train)

# Predict the classes for the test set

y\_pred = knn.predict(X\_test)

# Print both correct and wrong predictions

print(f"{'Index':<5} {'True Label':<10} {'Predicted':<10} {'Result'}")

print('-' \* 40)

for idx, (true, pred) in enumerate(zip(y\_test, y\_pred)):

result = "Correct" if true == pred else "Wrong"

print(f"{idx:<5} {true:<10} {pred:<10} {result}")

# Optionally, print overall accuracy

print("\nAccuracy:", accuracy\_score(y\_test, y\_pred))