```
# LINEAR REGRESSION
#
# y = b1 * x + b0
#
# To find b0 and b1 there are 2 ways:
#
# 1. Karl's correlation coeff
# 2. Least square method
#
# Steps (using Least Square Method):
# 1. Find mean of X (x")
# 2. Find mean of y (y")
# 3. Find x - x" and y - y"
# 4. Find (x - x") ** 2
# 5. Find b1 using formula
# 6. Find b0 using formula
# 7. Find y using regression formula

import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression

coeff = []

def createDataSet(X, y):
    df = pd.DataFrame({
        "X" : X,
        "Y" : y
    })
    return df

def calculateXY(df):
    X_mean = df["X"].mean()
    df["X - X'"] = round(df["X"] - X_mean, 2)
    Y_mean = df["Y"].mean()
    df["Y - Y'"] = round(df["Y"] - Y_mean, 2)
    df["(X - X') * (Y - Y')"] = df["X - X'"] * df["Y - Y'"]
    return df

def calculateXSquare(df):
    df["(X - X') ^ 2"] = df["X - X'"] ** 2
    return df

def calculatecoeff(x_val, df):
    xy_sum = df["(X - X') * (Y - Y')"].sum()
    xsquare_sum = df["(X - X') ^ 2"].sum()
    X_mean = df["X"].mean()
    Y_mean = df["Y"].mean()
    b1 = round(xy_sum / xsquare_sum, 2)
    b0 = round(Y_mean - b1 * X_mean, 2)
    coeff.append(b0)
    coeff.append(b1)
    print(coeff)
    print(x_val)
    y = round(coeff[0] + coeff[1] * x_val, 2)
    return y

def scikitlearn_linear_reg(x_val, df):
    X = df["X"].values
    X = np.reshape(X, (-1, 1))
    Y = df["Y"].values
    Y = np.reshape(Y, (-1, 1))
    reg = LinearRegression().fit(X, Y)
    y = round(reg.predict(np.array([[x_val]])).flatten()[0], 2)
    return y

if __name__ == "__main__":
    X = []
    y = []
    print("Enter dataset values for X: (Press q to quit)")
    while(True):
        n = input()
        if(n == 'q'):
            break
        else:
            X.append(int(n))
    print("Enter dataset values for Y: (Press q to quit)")
    while(True):
        n = input()
        if(n == 'q'):
            break
        else:
            y.append(int(n))
    df = createDataSet(X, y)
    x = int(input("Enter x value: "))
    df = calculateXY(df)
    df = calculateXSquare(df)
    y_custom = calculatecoeff(x, df)
    y_scikit = scikitlearn_linear_reg(x, df)
    print("FINAL DATA:")
    print(df.head(6))
    print(f"For x = {x}")
    print(f"By custom linear regression model: y = {y_custom}")
    print(f"By scikit-learn linear regression model: y = {y_scikit}")

########### OUTPUT ###########
# Enter dataset values for X: (Press q to quit)
# 0
# 1
# 2
# 3
# 4
# q
# Enter dataset values for Y: (Press q to quit)
# 2
# 3
# 5
# 4
# 6
# q
# Enter x value: 10
# [2.2, 0.9]
# 10
# FINAL DATA:
#    X  Y  X - X'  Y - Y'  (X - X') * (Y - Y')  (X - X') ^ 2
# 0  0  2    -2.0    -2.0                 4.0           4.0
# 1  1  3    -1.0    -1.0                 1.0           1.0
# 2  2  5     0.0     1.0                 0.0           0.0
# 3  3  4     1.0     0.0                 0.0           1.0
# 4  4  6     2.0     2.0                 4.0           4.0
# For x = 10
# By custom linear regression model: y = 11.2
# By scikit-learn linear regression model: y = 11.2
```