Program - 1

AIM :- Python program to demonstrate the
different operators in python.

```python
# Arithmetic Operators.
x = 10
y = 5
print ("Arithmetic Operators:")
print (" x+y = ", x + y)
print (" x-y = ", x - y)
print (" x * y = ", x * y)
print (" x / y = ", x / y)
print (" x % y = ", x % y)
print (" x ** y = ", x ** y)
print (" x // y = ", x // y)


# Comparison Operators.
a = 7
b = 12
print ("Comparison Operator:")
print (" a == b:", a == b)
print (" a != b:", a != b)
print (" a > b:", a > b)
print (" a < b:", a < b)
```

```
print (" a>= b:", a>= b)
print (" a<= b:", a<=b)


# Logical Operators.
p = True
q = False.
print ("Logical Operators:")
print (" p and q:", p and q)
print (" p or q:", p or q)
print (" not p:", not p)


# Assignment Operators
x = 10
y = 5
print ("Assignment Operators:")
print (" x += ", x += y)
print (" x= ", x -= y)
print (" x= ", x *= y)
print (" x= ", x /= y)
print (" x= ", x %= y)
print (" x= ", x **= y)
print (" x= ", x //= y)
```

```python
# Membership Operators
fruits = ["apple", "orange", "banana"]
print("Membership Operators: ")
print("'apple' in fruits: ", apple in fruits)
print("'grape' not in fruits: ", grape in fruits)


# Identity Operators.
x = 10
y = 10
z = 5
print("Identity Operators: ")
print(" x is y : ", x is y)
print(" x is z : ", x is z)
print(" x is not z : ", x is not z).
```

Program -2

AIM :- Find the mean, median, mode, variance and standard deviation of a list.

```python
import statistics
data = [1, 2, 3, 4, 5, 6, 7, 8, 9]

# Mean
mean = statistics.mean (data)
print ("Mean: ", mean)

# Median
median = statistics.median (data)
print ("Median: ", median)

# Mode
mode = statistics.mode (data)
print ("Mode: ", mode)

# Variance
variance = statistics.variance (data)
print ("Variance: ", variance)

# Standard Deviation
std_deviation = statistics.stdev (data)
print ("Standard Deviation: ", std_deviation)
```

Program - 3

__AIM :-__ Implement the linear regression algorithm

```
import numpy as np
from sklearn.linear_model import LinearRegression

# Sample data
X = np.array([[1],[2],[3],[4],[5]])
Y = np.array([2,4,5,7,8])

# Create and fit the model
model = LinearRegression()
model.fit(X,Y)

# Predict the output
X_test = np.array([[6],[7],[8]])
Y_pred = model.predict(X_test)

# Coefficient & Intercept.
print("Coefficients: ", model.coef_)
print("Intercepts: ", model.intercept_)

# Print predictions
print("Predictions: ", Y_pred).
```

Program-4

<u>AIM:-</u> Implement the Gradient Descent algourthm.

```
import numpy as np
def gradient_descent(x, y):
    m = c = 0
    rate = 0.001
    iteration = 100
    n = len(x)
    for i in range(iteration):
        y_predicted = m * x + c;
        cost = (1/n) * sum([val**2 for val in y-y_predicted])
        mp = -(2/n) * sum(x * (y-y_predicted))
        cp = -(2/n) * sum(y-y_predicted);
        m = m - rate * mp
        c = c - rate * cp

x = np.array([2,4,6,8,10,12,14,16,18,20])
y = np.array([10,20,30,40,50,60,70,80,90,100])
gradient_descent(x, y)

print('m:', m)
print('c:', c)
print('cost:', cost)
print('/n')
```