BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400058-India
Department of Computer Engineering

Name	Sujal Sandeep Dingankar
UID no.	2024301005
Experiment No.	3

AIM:	Learning NumPy Library
Program 1	
PROBLEM STATEMENT :	You are a data scientist working for a company that monitors environmental conditions using multiple sensors. Each sensor records temperature data throughout the day. You have the following tasks: sensor_1_data = [22, 21, 23, 24, 25, 24, 22] sensor_2_data = [20, 19, 21, 22, 23, 22, 21] sensor_3_data = [23, 22, 24, 25, 26, 25, 24] a) Convert these lists into NumPy arrays. b) Suppose you need to initialize an array to keep track of anomalies in thetemperature data. Create an array of zeros with the same shape as the temperature data for the day, and an array of ones with the same shape to represent ideal conditions. c) Calculate the daily average temperature for each sensor by summing upthe temperature data and dividing by the number of readings. d) If you find that the reading from the first sensor at 3rd value value is incorrect and should be 26 instead of 23, update the array accordingly. Reshape the temperature data from all three sensors into a 3x7 array, where each row represents one sensor's data. e) You want to adjust the temperature readings by subtracting a constant offset (e.g., 2 degrees) to correct for a known calibration issue. Use broadcasting to apply this offset to all sensor data. f) Apply the sine function to each temperature reading to simulate a transformation. g) Calculate the total sum and mean of the adjusted temperature readings.

PROGRAM:

```
import numpy as np

# Sensor data lists
sensor_1_data = [22, 21, 23, 24, 25, 24, 22]
sensor_2_data = [20, 19, 21, 22, 23, 22, 21]
sensor_3_data = [23, 22, 24, 25, 26, 25, 24]

# Convert these lists into NumPy arrays
data1 = np.array(sensor_1_data)
data2 = np.array(sensor_2_data)
data3 = np.array(sensor_3_data)

# Create arrays of zeroes and ones
zerosArr = np.zeros(np.shape(data1))
onesArr = np.ones(np.shape(data1))

# Calculate the daily average temperature for each sensor
sum1 = np.sum(data1)
print("Average daily temperature for sensor 1 is:", sum1 / len(data1))

sum2 = np.sum(data2)
print("Average daily temperature for sensor 2 is:", sum2 / len(data2))

sum3 = np.sum(data3)
print("Average daily temperature for sensor 3 is:", sum3 / len(data3))

# Correcting the data for sensor 1
data1[2] = 26
print("Corrected data for sensor 1, third entry:", data1[2])

# Reshaping the temperature data by concatenating all arrays
temp = np.concatenate((data1, data2, data3), axis=0)
Ans = temp.reshape(3, 7)
print("Reshaped array is:\n", Ans)

# Adjusting the readings by an offset
offset = 2
Ans = Ans - offset
```

```
print("Array after applying offset:\n", Ans)

# Apply the sine function to each temperature reading
sine_func = np.sin(Ans)
print("Sine function of each temperature reading:\n", sine_func)

# Calculate the total sum and mean of the adjusted temperature readings
total_sum = np.sum(Ans)
mean_val = np.mean(Ans)

print("Sum and mean of the adjusted temperature readings are:", total_sum,
"and", mean_val)
```

RESULT:

```
Average daily temperature for sensor 1 is: 23.0
Average daily temperature for sensor 2 is: 21.142857142857142
Average daily temperature for sensor 3 is: 24.142857142857142
Corrected data for sensor 1, third entry: 26
Reshaped array is:
 [[22 21 26 24 25 24 22]
 [20 19 21 22 23 22 21]
 [23 22 24 25 26 25 24]]
Array after applying offset:
 [[20 19 24 22 23 22 20]
 [18 17 19 20 21 20 19]
 [21 20 22 23 24 23 22]]
Sine function of each temperature reading:
 [[ 0.91294525  0.14987721 -0.90557836 -0.00885131 -0.8462204  -0.00885131
   0.91294525]
 [-0.75098725 -0.96139749  0.14987721  0.91294525  0.83665564  0.91294525
   0.14987721]
 [ 0.83665564  0.91294525 -0.00885131 -0.8462204  -0.90557836 -0.8462204
  -0.00885131]]
Sum and mean of the adjusted temperature readings are: 439 and 20.904761904761905
```

BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400058-India
Department of Computer Engineering

Program 2	
PROBLEM STATEMENT :	There a data set(NumPy.csv) with 10000 records, a) Create around 100 arrays of size 10x10 arrays. b) Multiply these arrays, till you get a single 10x10 array as an answer. c) Create a soft and deep copy of resultant array. d) Multiply the original array with 10. And print soft and hard copied array. e) Flatten and ravel the resultant array and pront them.

PROGRAM:	from google.colab import files
	uploaded = files.upload()
	
	file_path = "/content/NumPy.csv"
	data = np.genfromtxt(file_path, skip_header=1)
	
	result = []
	arr = data.shape[0] // 100
	
	# Splitting data into chunks of 100 rows and reshaping into 10x10 arrays
	for i in range(arr):
	  start = i * 100
	  end = (i + 1) * 100
	  array = data[start:end].reshape(10, 10)
	  result.append(array)
	
	# Multiply these arrays until you get a single 10x10 array as a result
	ans = np.ones((10, 10))
	for i in range(len(result)):
	  ans = ans * result[i]
	
	print(f"After Multiplication:\n{ans}")
	
	# Create a soft and deep copy of the resultant array
	soft = ans.view()
	print(f"Soft Copy:\n{soft}")

```
deep = ans.copy()
print(f"Deep Copy:\n{deep}")

# Multiply the original array by 10 and print soft and hard copies
ans *= 10
print(f"Soft Copy after changes:\n{soft}")
print(f"Deep Copy after changes:\n{deep}")

# Flatten and ravel the resultant array and print them
flatten = ans.flatten()
print(f"Flatten:\n{flatten}")

ravel = ans.ravel()
print(f"Ravel:\n{ravel}")
```

RESULT:

a) After Multiplication :-

```
After Multiplication:
[[2.42774298e+269 7.47271093e+268 2.16181160e+270 2.95563917e+270
  4.15845814e+269 7.72063595e+268 4.09808577e+269 6.49475875e+269
  5.54423048e+269 6.45535564e+271]
 [1.30615101e+270 2.01304993e+270 1.00441269e+271 3.02987424e+268
  2.47999921e+270 1.07151194e+270 2.02101845e+271 6.28866518e+268
  9.61582825e+269 2.21207055e+269]
 [1.25972898e+271 2.24242040e+270 2.22493576e+270 2.20510129e+271
  8.72038854e+269 9.48343895e+268 1.51764444e+269 2.16695628e+272
  5.30259314e+269 1.16615413e+270]
 [8.65987153e+269 4.39593834e+268 8.37841097e+270 7.85406676e+269
  1.97447875e+271 4.04993791e+270 2.90124443e+270 7.72371917e+268
  1.18650330e+270 5.49241848e+269]
 [1.19554687e+269 1.34918679e+271 2.41237723e+269 1.04817841e+269
  2.51637062e+270 8.04896616e+268 1.47430054e+268 1.24756442e+269
  6.74470804e+269 2.55261929e+269]
 [3.94884986e+269 1.81634643e+271 2.57407504e+269 6.47854724e+269
  1.20897868e+269 3.36740605e+269 7.70257633e+269 3.16384568e+271
  5.59319414e+269 2.61443252e+271]
 [8.35034786e+270 1.02699613e+271 1.04639192e+269 6.61291325e+268
  1.02073844e+271 5.65779228e+269 3.37426148e+271 1.70355867e+270
  1.19224704e+270 2.18096002e+271]
 [2.02043565e+270 1.43216091e+270 1.25418519e+268 2.95691518e+268
  4.53300055e+269 1.78183539e+270 1.41476325e+269 4.96651731e+269
  1.48340775e+270 4.38852796e+269]
 [1.59018655e+270 3.99870747e+270 2.43319925e+269 5.14223153e+270
  7.99168466e+269 1.10568960e+270 3.73362585e+269 9.18412622e+269
  3.12678664e+269 8.10213960e+270]
 [6.02687129e+269 8.36670654e+269 2.60429329e+270 6.43940621e+269
  1.26251504e+269 2.70762149e+270 2.21010632e+271 1.82879102e+270
  4.85127900e+269 4.63514373e+270]]
```

b) Soft Copy :-

```
Soft Copy:
[[2.42774298e+269 7.47271093e+268 2.16181160e+270 2.95563917e+270
  4.15845814e+269 7.72063595e+268 4.09808577e+269 6.49475875e+269
  5.54423048e+269 6.45535564e+271]
 [1.30615101e+270 2.01304993e+270 1.00441269e+271 3.02987424e+268
  2.47999921e+270 1.07151194e+270 2.02101845e+271 6.28866518e+268
  9.61582825e+269 2.21207055e+269]
 [1.25972898e+271 2.24242040e+270 2.22493576e+270 2.20510129e+271
  8.72038854e+269 9.48343895e+268 1.51764444e+269 2.16695628e+272
  5.30259314e+269 1.16615413e+270]
 [8.65987153e+269 4.39593834e+268 8.37841097e+270 7.85406676e+269
  1.97447875e+271 4.04993791e+270 2.90124443e+270 7.72371917e+268
  1.18650330e+270 5.49241848e+269]
 [1.19554687e+269 1.34918679e+271 2.41237723e+269 1.04817841e+269
  2.51637062e+270 8.04896616e+268 1.47430054e+268 1.24756442e+269
  6.74470804e+269 2.55261929e+269]
 [3.94884986e+269 1.81634643e+271 2.57407504e+269 6.47854724e+269
  1.20897868e+269 3.36740605e+269 7.70257633e+269 3.16384568e+271
  5.59319414e+269 2.61443252e+271]
 [8.35034786e+270 1.02699613e+271 1.04639192e+269 6.61291325e+268
  1.02073844e+271 5.65779228e+269 3.37426148e+271 1.70355867e+270
  1.19224704e+270 2.18096002e+271]
 [2.02043565e+270 1.43216091e+270 1.25418519e+268 2.95691518e+268
  4.53300055e+269 1.78183539e+270 1.41476325e+269 4.96651731e+269
  1.48340775e+270 4.38852796e+269]
 [1.59018655e+270 3.99870747e+270 2.43319925e+269 5.14223153e+270
  7.99168466e+269 1.10568960e+270 3.73362585e+269 9.18412622e+269
  3.12678664e+269 8.10213960e+270]
 [6.02687129e+269 8.36670654e+269 2.60429329e+270 6.43940621e+269
  1.26251504e+269 2.70762149e+270 2.21010632e+271 1.82879102e+270
  4.85127900e+269 4.63514373e+270]]
```

c) Deep Copy: :-

```
Deep Copy:
[[2.42774298e+269 7.47271093e+268 2.16181160e+270 2.95563917e+270
  4.15845814e+269 7.72063595e+268 4.09808577e+269 6.49475875e+269
  5.54423048e+269 6.45535564e+271]
 [1.30615101e+270 2.01304993e+270 1.00441269e+271 3.02987424e+268
  2.47999921e+270 1.07151194e+270 2.02101845e+271 6.28866518e+268
  9.61582825e+269 2.21207055e+269]
 [1.25972898e+271 2.24242040e+270 2.22493576e+270 2.20510129e+271
  8.72038854e+269 9.48343895e+268 1.51764444e+269 2.16695628e+272
  5.30259314e+269 1.16615413e+270]
 [8.65987153e+269 4.39593834e+268 8.37841097e+270 7.85406676e+269
  1.97447875e+271 4.04993791e+270 2.90124443e+270 7.72371917e+268
  1.18650330e+270 5.49241848e+269]
 [1.19554687e+269 1.34918679e+271 2.41237723e+269 1.04817841e+269
  2.51637062e+270 8.04896616e+268 1.47430054e+268 1.24756442e+269
  6.74470804e+269 2.55261929e+269]
 [3.94884986e+269 1.81634643e+271 2.57407504e+269 6.47854724e+269
  1.20897868e+269 3.36740605e+269 7.70257633e+269 3.16384568e+271
  5.59319414e+269 2.61443252e+271]
 [8.35034786e+270 1.02699613e+271 1.04639192e+269 6.61291325e+268
  1.02073844e+271 5.65779228e+269 3.37426148e+271 1.70355867e+270
  1.19224704e+270 2.18096002e+271]
 [2.02043565e+270 1.43216091e+270 1.25418519e+268 2.95691518e+268
  4.53300055e+269 1.78183539e+270 1.41476325e+269 4.96651731e+269
  1.48340775e+270 4.38852796e+269]
 [1.59018655e+270 3.99870747e+270 2.43319925e+269 5.14223153e+270
  7.99168466e+269 1.10568960e+270 3.73362585e+269 9.18412622e+269
  3.12678664e+269 8.10213960e+270]
 [6.02687129e+269 8.36670654e+269 2.60429329e+270 6.43940621e+269
  1.26251504e+269 2.70762149e+270 2.21010632e+271 1.82879102e+270
  4.85127900e+269 4.63514373e+270]]
```

d) Soft Copy after changes :-

```
Soft Copy after changes:
[[2.42774298e+270 7.47271093e+269 2.16181160e+271 2.95563917e+271
  4.15845814e+270 7.72063595e+269 4.09808577e+270 6.49475875e+270
  5.54423048e+270 6.45535564e+272]
 [1.30615101e+271 2.01304993e+271 1.00441269e+272 3.02987424e+269
  2.47999921e+271 1.07151194e+271 2.02101845e+272 6.28866518e+269
  9.61582825e+270 2.21207055e+270]
 [1.25972898e+272 2.24242040e+271 2.22493576e+271 2.20510129e+272
  8.72038854e+270 9.48343895e+269 1.51764444e+270 2.16695628e+273
  5.30259314e+270 1.16615413e+271]
 [8.65987153e+270 4.39593834e+269 8.37841097e+271 7.85406676e+270
  1.97447875e+272 4.04993791e+271 2.90124443e+271 7.72371917e+269
  1.18650330e+271 5.49241848e+270]
 [1.19554687e+270 1.34918679e+272 2.41237723e+270 1.04817841e+270
  2.51637062e+271 8.04896616e+269 1.47430054e+269 1.24756442e+270
  6.74470804e+270 2.55261929e+270]
 [3.94884986e+270 1.81634643e+272 2.57407504e+270 6.47854724e+270
  1.20897868e+270 3.36740605e+270 7.70257633e+270 3.16384568e+272
  5.59319414e+270 2.61443252e+272]
 [8.35034786e+271 1.02699613e+272 1.04639192e+270 6.61291325e+269
  1.02073844e+272 5.65779228e+270 3.37426148e+272 1.70355867e+271
  1.19224704e+271 2.18096002e+272]
 [2.02043565e+271 1.43216091e+271 1.25418519e+269 2.95691518e+269
  4.53300055e+270 1.78183539e+271 1.41476325e+270 4.96651731e+270
  1.48340775e+271 4.38852796e+270]
 [1.59018655e+271 3.99870747e+271 2.43319925e+270 5.14223153e+271
  7.99168466e+270 1.10568960e+271 3.73362585e+270 9.18412622e+270
  3.12678664e+270 8.10213960e+271]
 [6.02687129e+270 8.36670654e+270 2.60429329e+271 6.43940621e+270
  1.26251504e+270 2.70762149e+271 2.21010632e+272 1.82879102e+271
  4.85127900e+270 4.63514373e+271]]
```

e) Deep Copy after changes :-

```
Deep Copy after changes:
[[2.42774298e+269 7.47271093e+268 2.16181160e+270 2.95563917e+270
  4.15845814e+269 7.72063595e+268 4.09808577e+269 6.49475875e+269
  5.54423048e+269 6.45535564e+271]
 [1.30615101e+270 2.01304993e+270 1.00441269e+271 3.02987424e+268
  2.47999921e+270 1.07151194e+270 2.02101845e+271 6.28866518e+268
  9.61582825e+269 2.21207055e+269]
 [1.25972898e+271 2.24242040e+270 2.22493576e+270 2.20510129e+271
  8.72038854e+269 9.48343895e+268 1.51764444e+269 2.16695628e+272
  5.30259314e+269 1.16615413e+270]
 [8.65987153e+269 4.39593834e+268 8.37841097e+270 7.85406676e+269
  1.97447875e+271 4.04993791e+270 2.90124443e+270 7.72371917e+268
  1.18650330e+270 5.49241848e+269]
 [1.19554687e+269 1.34918679e+271 2.41237723e+269 1.04817841e+269
  2.51637062e+270 8.04896616e+268 1.47430054e+268 1.24756442e+269
  6.74470804e+269 2.55261929e+269]
 [3.94884986e+269 1.81634643e+271 2.57407504e+269 6.47854724e+269
  1.20897868e+269 3.36740605e+269 7.70257633e+269 3.16384568e+271
  5.59319414e+269 2.61443252e+271]
 [8.35034786e+270 1.02699613e+271 1.04639192e+269 6.61291325e+268
  1.02073844e+271 5.65779228e+269 3.37426148e+271 1.70355867e+270
  1.19224704e+270 2.18096002e+271]
 [2.02043565e+270 1.43216091e+270 1.25418519e+268 2.95691518e+268
  4.53300055e+269 1.78183539e+270 1.41476325e+269 4.96651731e+269
  1.48340775e+270 4.38852796e+269]
 [1.59018655e+270 3.99870747e+270 2.43319925e+269 5.14223153e+270
  7.99168466e+269 1.10568960e+270 3.73362585e+269 9.18412622e+269
  3.12678664e+269 8.10213960e+270]
 [6.02687129e+269 8.36670654e+269 2.60429329e+270 6.43940621e+269
  1.26251504e+269 2.70762149e+270 2.21010632e+271 1.82879102e+270
  4.85127900e+269 4.63514373e+270]]
```

f) Flatten :-

```
Flatten:
[2.42774298e+270 7.47271093e+269 2.16181160e+271 2.95563917e+271
 4.15845814e+270 7.72063595e+269 4.09808577e+270 6.49475875e+270
 5.54423048e+270 6.45535564e+272 1.30615101e+271 2.01304993e+271
 1.00441269e+272 3.02987424e+269 2.47999921e+271 1.07151194e+271
 2.02101845e+272 6.28866518e+269 9.61582825e+270 2.21207055e+270
 1.25972898e+272 2.24242040e+271 2.22493576e+271 2.20510129e+272
 8.72038854e+270 9.48343895e+269 1.51764444e+270 2.16695628e+273
 5.30259314e+270 1.16615413e+271 8.65987153e+270 4.39593834e+269
 8.37841097e+271 7.85406676e+270 1.97447875e+272 4.04993791e+271
 2.90124443e+271 7.72371917e+269 1.18650330e+271 5.49241848e+270
 1.19554687e+270 1.34918679e+272 2.41237723e+270 1.04817841e+270
 2.51637062e+271 8.04896616e+269 1.47430054e+269 1.24756442e+270
 6.74470804e+270 2.55261929e+270 3.94884986e+270 1.81634643e+272
 2.57407504e+270 6.47854724e+270 1.20897868e+270 3.36740605e+270
 7.70257633e+270 3.16384568e+272 5.59319414e+270 2.61443252e+272
 8.35034786e+271 1.02699613e+272 1.04639192e+270 6.61291325e+269
 1.02073844e+272 5.65779228e+270 3.37426148e+272 1.70355867e+271
 1.19224704e+271 2.18096002e+272 2.02043565e+271 1.43216091e+271
 1.25418519e+269 2.95691518e+269 4.53300055e+270 1.78183539e+271
 1.41476325e+270 4.96651731e+270 1.48340775e+271 4.38852796e+270
 1.59018655e+271 3.99870747e+271 2.43319925e+270 5.14223153e+271
 7.99168466e+270 1.10568960e+271 3.73362585e+270 9.18412622e+270
 3.12678664e+270 8.10213960e+271 6.02687129e+270 8.36670654e+270
 2.60429329e+271 6.43940621e+270 1.26251504e+270 2.70762149e+271
 2.21010632e+272 1.82879102e+271 4.85127900e+270 4.63514373e+271]
```

g) Ravel :-

```
Ravel:
[2.42774298e+270 7.47271093e+269 2.16181160e+271 2.95563917e+271
 4.15845814e+270 7.72063595e+269 4.09808577e+270 6.49475875e+270
 5.54423048e+270 6.45535564e+272 1.30615101e+271 2.01304993e+271
 1.00441269e+272 3.02987424e+269 2.47999921e+271 1.07151194e+271
 2.02101845e+272 6.28866518e+269 9.61582825e+270 2.21207055e+270
 1.25972898e+272 2.24242040e+271 2.22493576e+271 2.20510129e+272
 8.72038854e+270 9.48343895e+269 1.51764444e+270 2.16695628e+273
 5.30259314e+270 1.16615413e+271 8.65987153e+270 4.39593834e+269
 8.37841097e+271 7.85406676e+270 1.97447875e+272 4.04993791e+271
 2.90124443e+271 7.72371917e+269 1.18650330e+271 5.49241848e+270
 1.19554687e+270 1.34918679e+272 2.41237723e+270 1.04817841e+270
 2.51637062e+271 8.04896616e+269 1.47430054e+269 1.24756442e+270
 6.74470804e+270 2.55261929e+270 3.94884986e+270 1.81634643e+272
 2.57407504e+270 6.47854724e+270 1.20897868e+270 3.36740605e+270
 7.70257633e+270 3.16384568e+272 5.59319414e+270 2.61443252e+272
 8.35034786e+271 1.02699613e+272 1.04639192e+270 6.61291325e+269
 1.02073844e+272 5.65779228e+270 3.37426148e+272 1.70355867e+271
 1.19224704e+271 2.18096002e+272 2.02043565e+271 1.43216091e+271
 1.25418519e+269 2.95691518e+269 4.53300055e+270 1.78183539e+271
 1.41476325e+270 4.96651731e+270 1.48340775e+271 4.38852796e+270
 1.59018655e+271 3.99870747e+271 2.43319925e+270 5.14223153e+271
 7.99168466e+270 1.10568960e+271 3.73362585e+270 9.18412622e+270
 3.12678664e+270 8.10213960e+271 6.02687129e+270 8.36670654e+270
 2.60429329e+271 6.43940621e+270 1.26251504e+270 2.70762149e+271
 2.21010632e+272 1.82879102e+271 4.85127900e+270 4.63514373e+271]
```

CONCLUSION:	In this experiment, I got acquainted with the NumPy library in python and learned many functions that you could perform on ndarrays.