Name:-Sakshi Y. Dube

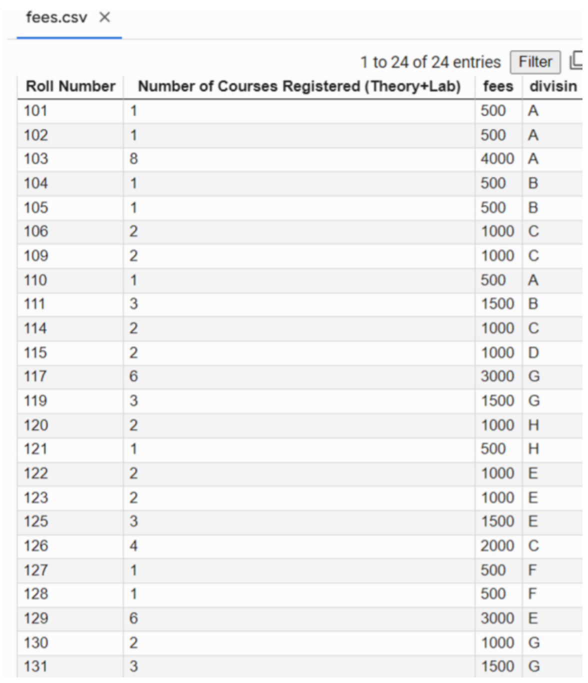
Roll no.:-721 Batch:-G1

Practical no.3

Problem statement:-

Prepare/Take datasets for any real-life application. Read a dataset into an array. Perform the following operations on it:

1. **Perform all matrix operations**
2. **Horizontal and vertical stacking of Numpy Arrays**
3. **Custom sequence generation**
4. **Arithmetic and Statistical Operations, Mathematical Operations, Bitwise Operators**
5. **Copying and viewing arrays**
6. **Data Stacking, Searching, Sorting, Counting, Broadcasting Csv File:-**



Roll Number	Number of Courses Registered (Theory+Lab)	fees	divisin
101	1	500	A
102	1	500	A
103	8	4000	A
104	1	500	B
105	1	500	B
106	2	1000	C
109	2	1000	C
110	1	500	A
111	3	1500	B
114	2	1000	C
115	2	1000	D
117	6	3000	G
119	3	1500	G
120	2	1000	H
121	1	500	H
122	2	1000	E
123	2	1000	E
125	3	1500	E
126	4	2000	C
127	1	500	F
128	1	500	F
129	6	3000	E
130	2	1000	G
131	3	1500	G

Program:-

1.

```
import numpy as np
array3=np.loadtxt('/content/fees.csv',delimiter=',',dtype=str,skiprows=1)
print(array3) roll_number=[] number_of_courses=[] fees=[] division=[] for
i in array3:
  roll_number.append(int(i[0]))
number_of_courses.append(int(i[1]))
fees.append(int(i[2]))   division.append((i[3]))
print(roll_number) print(number_of_courses)
print(fees) print(division)
arr_roll_number=np.array(roll_number)
arr_number_of_courses=np.array(number_of_courses)
arr_fees=np.array(fees)
arr_division=np.array(division)
 print('array1:',arr_roll_number)
print('array2:',arr_number_of_courses)
print('array3:',arr_fees)
print('array4:',arr_division)
```

```
print(np.min(arr_fees)) print(np.max(arr_fees))
print(np.sum(arr_fees))
print(np.sum(arr_fees/len(arr_fees)))
```

```
arr_number_of_courses*arr_fees arr_number_of_courses+arr_fees
arr_fees-arr_number_of_courses arr_fees/arr_number_of_courses
 2.
arr3=np.vstack((arr_number_of_courses,arr_fees))
print(arr3)
arr4=np.hstack((arr_number_of_courses,arr_fees)) print(arr4)
arr3=np.vstack((arr_number_of_courses,arr_roll_number))
print(arr3)
arr4=np.hstack((arr_number_of_courses,arr_roll_number))
print(arr4)
 3. indices =
np.arange(len(arr_number_of_courses)) for i in
indices:
    print("number of courses at index", i, ":", arr_number_of_courses[i])
4.
```

```
arr_number_of_courses*arr_fees
arr_number_of_courses+arr_fees arr_fees-
arr_number_of_courses
arr_fees/arr_number_of_courses
mx=np.array([arr_fees,arr_number_of_courses])
print("matrix=\n",mx) print("\nthe transpose")
print(mx.T)
  mean_a=np.mean(arr_fees)
print(mean_a)
median_a=np.median(arr_fees)
print(median_a)
std_a=np.std(arr_fees)
print(std_a)


arr1=np.bitwise_or(arr_roll_number,arr_fees) print(arr1)
arr2=np.bitwise_and(arr_roll_number,arr_fees) print(arr2)
arr3=np.bitwise_xor(arr_roll_number,arr_fees) print(arr3)
arr4=np.bitwise_not(arr_number_of_courses) print(arr4)
 5.
arr=arr_number_of_courses.copy() print(arr)
arr_roll_number.view()


6.
arr3=np.vstack((arr_number_of_courses,arr_fees)) print(arr3)
arr4=np.hstack((arr_number_of_courses,arr_fees)) print(arr4)
arr3=np.vstack((arr_number_of_courses,arr_roll_number))
print(arr3)
arr4=np.hstack((arr_number_of_courses,arr_roll_number))
print(arr4)
 print(arr_number_of_courses[1:5])
print(arr_fees[1:6])
print(arr_roll_number[1:6])
 arr_fees=np.arange(20) print("\n array
is:",arr_fees) print("\n arr_fees[-
8:17:1]=",arr_fees[-8:17:1]) print("\n
arr_fees[10:]=",arr_fees[10:])
 arr5=np.array(arr_fees)
print(np.sort(arr5))
print(np.sort(arr_fees))
print(np.sort(roll_number))
print(np.sort(number_of_courses))
 import numpy as np
np.count_nonzero(arr_fees==4000)
np.count_nonzero(arr_fees==500)
np.count_nonzero(arr_fees<1500)
 a=arr_fees[1:5] print(a)
b=arr_number_of_courses[1:5]
print(b) c=a+b print(c)
```

Output:-

```
1.
[['101' '1' '500' 'A' '']
 ['102' '1' '500' 'A' '']
 ['103' '8' '4000' 'A' '']
 ['104' '1' '500' 'B' '']
 ['105' '1' '500' 'B' '']
 ['106' '2' '1000' 'C' '']  ['109' '2' '1000' 'C' '']  ['110' '1' '500' 'A'
'']
 ['111' '3' '1500' 'B' '']
 ['114' '2' '1000' 'C' '']
 ['115' '2' '1000' 'D' '']
 ['117' '6' '3000' 'G' '']
 ['119' '3' '1500' 'G' '']
 ['120' '2' '1000' 'H' '']
['121' '1' '500' 'H' '']
 ['122' '2' '1000' 'E' '']
 ['123' '2' '1000' 'E' '']
 ['125' '3' '1500' 'E' '']
 ['126' '4' '2000' 'C' '']
 ['127' '1' '500' 'F' '']
 ['128' '1' '500' 'F' '']
 ['129' '6' '3000' 'E' '']
 ['130' '2' '1000' 'G' '']
 ['131' '3' '1500' 'G' '']]
[101, 102, 103, 104, 105, 106, 109, 110, 111, 114, 115, 117, 119, 120,
121, 122, 123, 125, 126, 127, 128, 129, 130, 131]
[1, 1, 8, 1, 1, 2, 2, 1, 3, 2, 2, 6, 3, 2, 1, 2, 2, 3, 4, 1, 1, 6, 2, 3]
[500, 500, 4000, 500, 500, 1000, 1000, 500, 1500, 1000, 1000, 3000, 1500,
1000, 500, 1000, 1000, 1500, 2000, 500, 500, 3000, 1000, 1500]
['A', 'A', 'A', 'B', 'B', 'C', 'C', 'A', 'B', 'C', 'D', 'G', 'G', 'H',
'H', 'E', 'E', 'E', 'C', 'F', 'F', 'E', 'G', 'G']
array1: [101 102 103 104 105 106 109 110 111 114 115 117 119 120 121 122
123 125
 126 127 128 129 130 131]
array2: [1 1 8 1 1 2 2 1 3 2 2 6 3 2 1 2 2 3 4 1 1 6 2 3]
array3: [ 500  500 4000  500  500 1000 1000  500 1500 1000 1000 3000 1500
1000
```

```
  500 1000 1000 1500 2000  500  500 3000 1000 1500]
array4: ['A' 'A' 'A' 'B' 'B' 'C' 'C' 'A' 'B' 'C' 'D' 'G' 'G' 'H' 'H' 'E'
'E' 'E'
 'C' 'F' 'F' 'E' 'G' 'G']
```

```
500
4000
30000 1250.0
```

```
array([ 500, 500, 32000, 500, 500, 2000, 2000, 500, 4500, 2000, 2000,
18000, 4500, 2000, 500, 2000, 2000, 4500, 8000, 500, 500, 18000, 2000,
4500])
```

```
array([ 501, 501, 4008, 501, 501, 1002, 1002, 501, 1503, 1002, 1002, 3006,
1503, 1002, 501, 1002, 1002, 1503, 2004, 501, 501, 3006, 1002, 1503])
array([ 499, 499, 3992, 499, 499, 998, 998, 499, 1497, 998, 998, 2994,
1497, 998, 499, 998, 998, 1497, 1996, 499, 499, 2994, 998, 1497])
array([500., 500., 500., 500., 500., 500., 500., 500., 500., 500., 500.,
500., 500., 500., 500., 500., 500., 500., 500., 500., 500., 500., 500.,
500.])
```

2.

```
[[   1    1    8    1    1    2    2    1    3    2    2    6    3    2
1    2    2    3    4    1    1    6    2    3]
 [ 500  500 4000  500  500 1000 1000  500 1500 1000 1000 3000 1500 1000
   500 1000 1000 1500 2000  500  500 3000 1000 1500]]
[   1    1    8    1    1    2    2    1    3    2    2    6    3    2
    1    2    2    3    4    1    1    6    2    3  500  500 4000  500
  500 1000 1000  500 1500 1000 1000 3000 1500 1000  500 1000 1000 1500
 2000  500  500 3000 1000 1500]
[[  1   1   8   1   1   2   2   1   3   2   2   6   3   2   1   2   2   3
4   1   1   6   2   3]
 [101 102 103 104 105 106 109 110 111 114 115 117 119 120 121 122 123 125
  126 127 128 129 130 131]]
[  1   1   8   1   1   2   2   1   3   2   2   6   3   2   1   2   2   3
   4   1   1   6   2   3 101 102 103 104 105 106 109 110 111 114 115 117
 119 120 121 122 123 125 126 127 128 129 130 131]
```

3.

```
number of courses at index 0 : 1 number
of courses at index 1 : 1 number of
courses at index 2 : 8 number of
courses at index 3 : 1 number of
courses at index 4 : 1 number of
courses at index 5 : 2 number of
courses at index 6 : 2 number of
courses at index 7 : 1 number of
courses at index 8 : 3 number of
courses at index 9 : 2 number of
courses at index 10 : 2 number of
courses at index 11 : 6 number of
courses at index 12 : 3 number of
courses at index 13 : 2 number of
courses at index 14 : 1 number of
courses at index 15 : 2 number of
courses at index 16 : 2 number of
courses at index 17 : 3 number of
courses at index 18 : 4 number of
courses at index 19 : 1 number of
courses at index 20 : 1 number of
courses at index 21 : 6 number of
courses at index 22 : 2 number of
courses at index 23 : 3
```

4.

```
array([ 500, 500, 32000, 500, 500, 2000, 2000, 500, 4500, 2000, 2000,
18000, 4500, 2000, 500, 2000, 2000, 4500, 8000, 500, 500, 18000, 2000,
4500])
array([ 501, 501, 4008, 501, 501, 1002, 1002, 501, 1503, 1002, 1002, 3006,
1503, 1002, 501, 1002, 1002, 1503, 2004, 501, 501, 3006, 1002, 1503])
array([ 499, 499, 3992, 499, 499, 998, 998, 499, 1497, 998, 998, 2994,
1497, 998, 499, 998, 998, 1497, 1996, 499, 499, 2994, 998, 1497])
array([500., 500., 500., 500., 500., 500., 500., 500., 500., 500., 500.,
500., 500., 500., 500., 500., 500., 500., 500., 500., 500., 500., 500.,
500.])

matrix=
 [array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
16,
        17, 18, 19])
 array([1, 1, 8, 1, 1, 2, 2, 1, 3, 2, 2, 6, 3, 2, 1, 2, 2, 3, 4, 1, 1, 6,
2, 3])                                                            ]

the transpose
[array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
16,
        17, 18, 19])
 array([1, 1, 8, 1, 1, 2, 2, 1, 3, 2, 2, 6, 3, 2, 1, 2, 2, 3, 4, 1, 1, 6,
2, 3])

1250.0

1000.0

901.3878188659974
```

```
[ 501  502 4071  508  509 1002 1005  510 1535 1018 1019 3069 1535 1016
  509 1018 1019 1533 2046  511  500 3001 1002 1503]
[100 100  32  96  96 104 104 100  76  96  96  48  84 104 112 104 104  92
  80 116 128 128 128 128]
[ 401  402 4039  412  413  898  901  410 1459  922  923 3021 1451  912
  397  914  915 1441 1966  395  372 2873  874 1375]
[-2 -2 -9 -2 -2 -3 -3 -2 -4 -3 -3 -7 -4 -3 -2 -3 -3 -4 -5 -2 -2 -7 -3 -4]
```

5.

```
[1 1 8 1 1 2 2 1 3 2 2 6 3 2 1 2 2 3 4 1 1 6 2 3]

array([101, 102, 103, 104, 105, 106, 109, 110, 111, 114, 115, 117, 119,
120, 121, 122, 123, 125, 126, 127, 128, 129, 130, 131])
```

6.

```
[[   1    1    8    1    1    2    2    1    3    2    2    6    3    2
1    2    2    3    4    1    1    6    2    3]
 [ 500  500 4000  500  500 1000 1000  500 1500 1000 1000 3000 1500 1000
   500 1000 1000 1500 2000  500  500 3000 1000 1500]]
[   1    1    8    1    1    2    2    1    3    2    2    6    3    2
    1    2    2    3    4    1    1    6    2    3  500  500 4000  500
  500 1000 1000  500 1500 1000 1000 3000 1500 1000  500 1000 1000 1500
2000  500  500 3000 1000 1500]
[[  1   1   8   1   1   2   2   1   3   2   2   6   3   2   1   2   2   3
4   1   1   6   2   3]
 [101 102 103 104 105 106 109 110 111 114 115 117 119 120 121 122 123 125
  126 127 128 129 130 131]]
[  1   1   8   1   1   2   2   1   3   2   2   6   3   2   1   2   2   3
   4   1   1   6   2   3 101 102 103 104 105 106 109 110 111 114 115 117
 119 120 121 122 123 125 126 127 128 129 130 131]

[1 8 1 1]
[ 500 4000  500  500 1000] [102
103 104 105 106]

array is: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]

 arr_fees[-8:17:1]= [12 13 14 15 16]

 arr_fees[10:]= [10 11 12 13 14 15 16 17 18 19]

[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]

[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
[101 102 103 104 105 106 109 110 111 114 115 117 119 120 121 122 123 125
 126 127 128 129 130 131]
[1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 3 4 6 6 8]
```

```
1

8

16

[ 500 4000  500  500]
[1 8 1 1]
[ 501 4008  501  501]
```