Name: D R YUVAM

College Roll No.: CSC/20/48

PROGRAM 1:

Array And Metrices

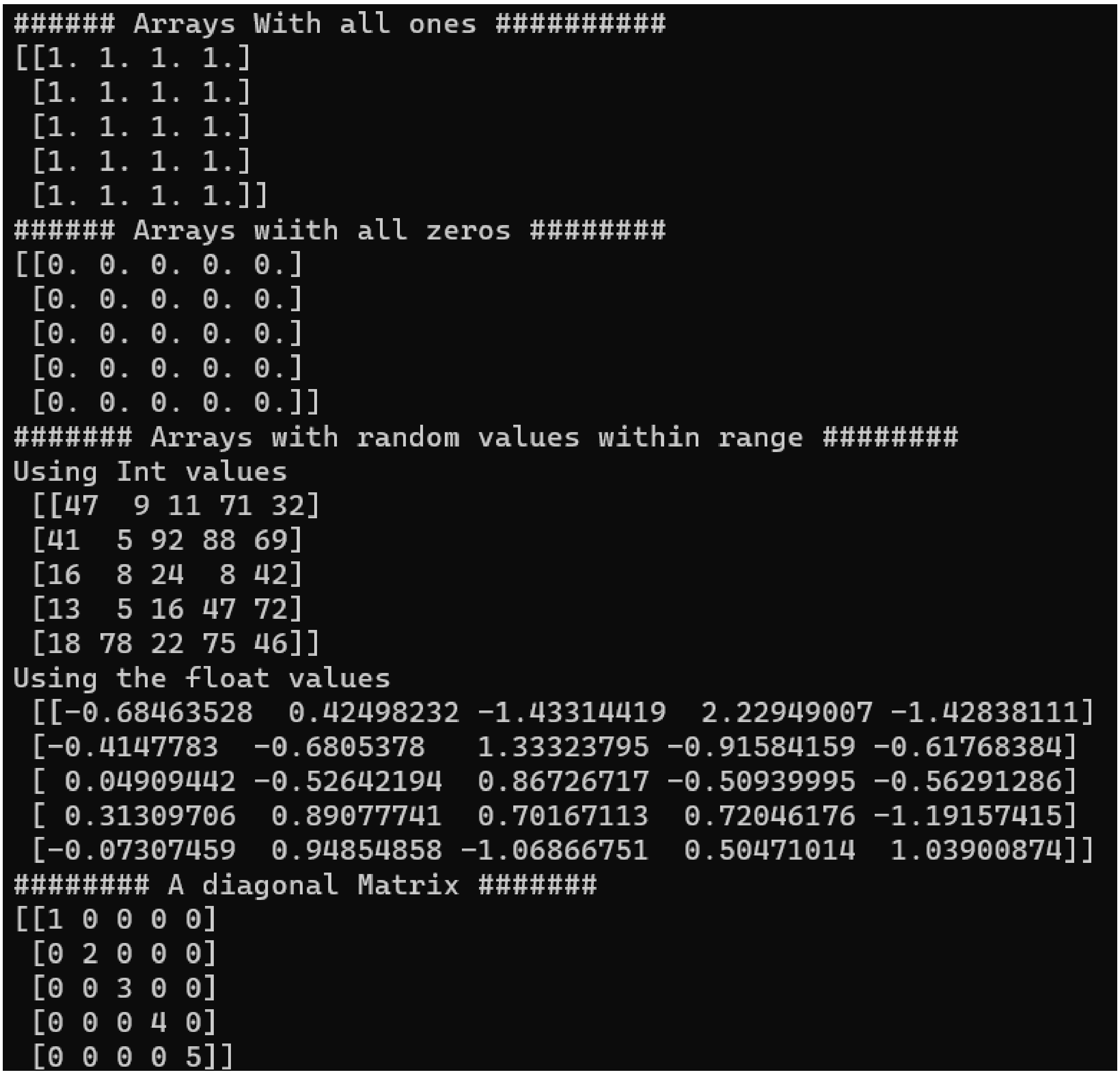
Ques 1:
(a) Create arrays with specific values:

(i) array of all ones
(ii) array of all zeros
(iii) array with random values within a range
(iv) a diagonal matrix

SOLUTION:

```
import numpy as np

def createArrays():
    print("###### Arrays With all ones ##########")
    print(np.ones(shape=(5,4)))

    print("###### Arrays wiith all zeros ########")
    print(np.zeros(shape=(5,5)))

    print("####### Arrays with random values within range
########")
    # using int values in the np random randint fucntion 1=lower
range , 100= higher range
    print("Using Int
values\n",np.random.randint(1,100,size=(5,5)))

    print("Using the float values\n",np.random.randn(5,5))

    print("######## A diagonal Matrix #######")
    # data for the diagonal of the diagonal matrix
    data = [1,2,3,4,5]
    print(np.diag(v=data))
    # v stands for vector it can be 1d or 2d data

if __name__ == "__main__":
    # Function to create arrays
    createArrays()
```

OUTPUT



```
###### Arrays With all ones ##########
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
###### Arrays wiith all zeros ########
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
####### Arrays with random values within range ########
Using Int values
 [[47  9 11 71 32]
 [41  5 92 88 69]
 [16  8 24  8 42]
 [13  5 16 47 72]
 [18 78 22 75 46]]
Using the float values
 [[-0.68463528  0.42498232 -1.43314419  2.22949007 -1.42838111]
 [-0.4147783  -0.6805378   1.33323795 -0.91584159 -0.61768384]
 [ 0.04909442 -0.52642194  0.86726717 -0.50939995 -0.56291286]
 [ 0.31309706  0.89077741  0.70167113  0.72046176 -1.19157415]
 [-0.07307459  0.94854858 -1.06866751  0.50471014  1.03900874]]
######## A diagonal Matrix #######
[[1 0 0 0 0]
 [0 2 0 0 0]
 [0 0 3 0 0]
 [0 0 0 4 0]
 [0 0 0 0 5]]
```

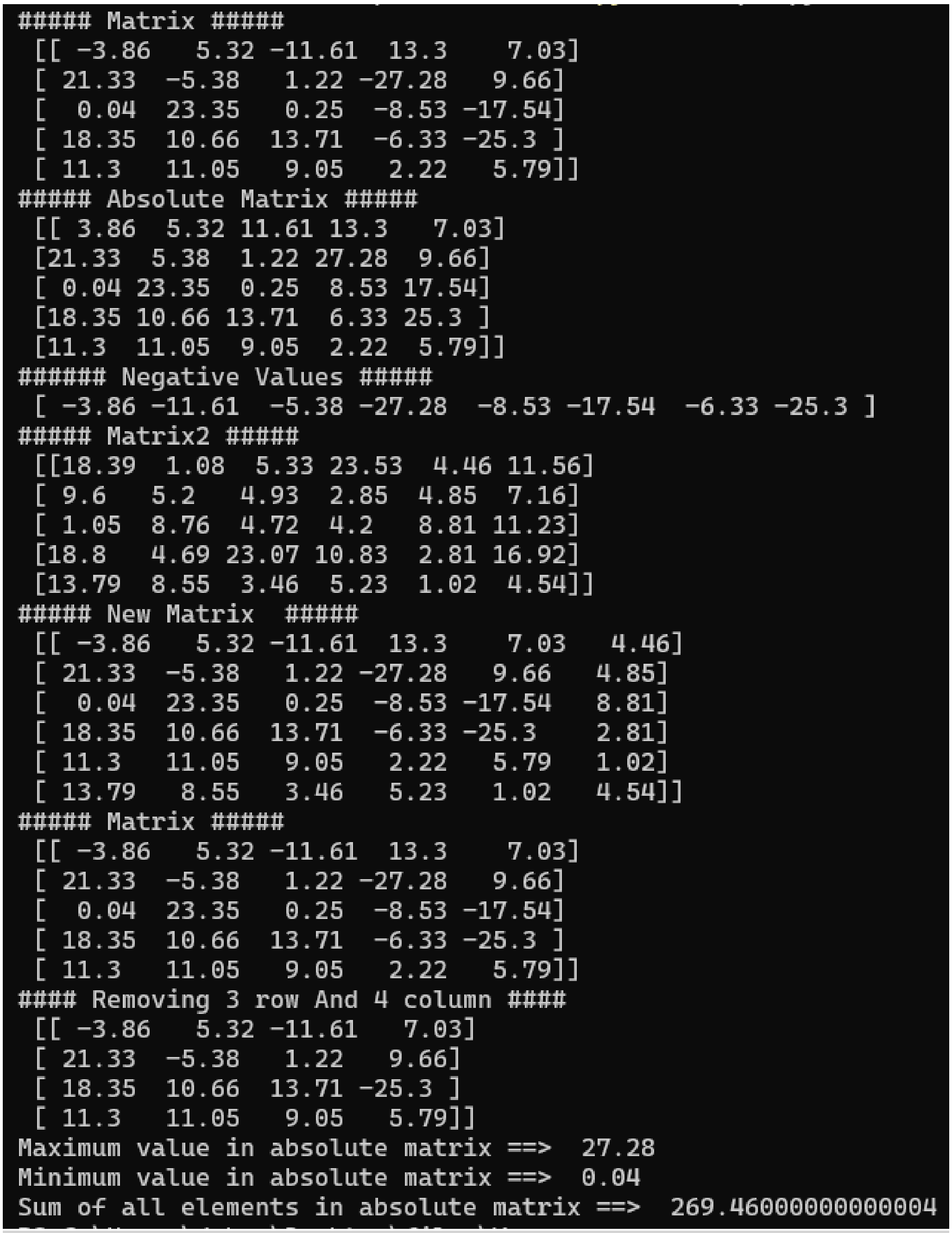
b) Perform other matrix operations:

(i) convert matrix data to absolute values
(ii) take the negative of matrix values
(iii) add rows & columns from a matrix
(iv) remove rows & columns from a matrix
(v) find the maximum and minimum values in a matrix or in a row/column
(vi) find the sum of some/all elements in a matrix

SOLUTION :

```python
import numpy as np

def manipulateMatrix(matrix):
    print("##### Absolute Matrix #####\n",np.abs(matrix))

    print("###### Negative Values #####\n",matrix[np.where(matrix<0)])

    addColAndRow(matrix)
    removeColAndRow(matrix,2,3)
    matrix = np.abs(matrix)
    maximum,minimum = np.max(matrix),np.min(matrix)
    print("Maximum value in absolute matrix ==> ",maximum)
    print("Minimum value in absolute matrix ==> ",minimum)

    print("Sum of all elements in absolute matrix ==> ",np.sum(matrix))

def addColAndRow(matrix):
    matrix2 = np.abs(np.round_(np.random.randn(5,6) * 10,decimals=2))
    print("##### Matrix2 #####\n",matrix2)

    x1,y1 = np.shape(matrix)
    x2,y2 = np.shape(matrix2)

    # adding a column using insert method
    newmatrix = np.insert(matrix,y1,matrix2[:,x2-1],axis=1)

    # adding a rows using append method
    newmatrix = np.vstack((newmatrix,matrix2[x1-1]))

    # note last rows and columns are added to the matrix

    print("##### New Matrix  #####\n",newmatrix)

def removeColAndRow(matrix,row,col):
    print("##### Matrix #####\n",matrix)
    newmatrix2 = np.delete(matrix,row,axis=0)
    newmatrix2 = np.delete(newmatrix2,col,axis=1)
    print(f"#### Removing {row+1} row And {col+1} column ####\n",newmatrix2)


if __name__ == "__main__":
    # created a sample matrix rrounding off to 2 decimal places and then multiplying the values by 10
    matrix = np.round_(np.random.randn(5,5) * 10,decimals=2)
    print("##### Matrix #####\n",matrix)
    manipulateMatrix(matrix)
```

OUTPUT



```
##### Matrix #####
 [[ -3.86   5.32 -11.61  13.3    7.03]
 [ 21.33  -5.38   1.22 -27.28   9.66]
 [  0.04  23.35   0.25  -8.53 -17.54]
 [ 18.35  10.66  13.71  -6.33 -25.3 ]
 [ 11.3   11.05   9.05   2.22   5.79]]
##### Absolute Matrix #####
 [[ 3.86  5.32 11.61 13.3   7.03]
 [21.33  5.38  1.22 27.28  9.66]
 [ 0.04 23.35  0.25  8.53 17.54]
 [18.35 10.66 13.71  6.33 25.3 ]
 [11.3  11.05  9.05  2.22  5.79]]
###### Negative Values #####
 [ -3.86 -11.61  -5.38 -27.28  -8.53 -17.54  -6.33 -25.3 ]
##### Matrix2 #####
 [[18.39  1.08  5.33 23.53  4.46 11.56]
 [ 9.6   5.2   4.93  2.85  4.85  7.16]
 [ 1.05  8.76  4.72  4.2   8.81 11.23]
 [18.8   4.69 23.07 10.83  2.81 16.92]
 [13.79  8.55  3.46  5.23  1.02  4.54]]
##### New Matrix  #####
 [[ -3.86   5.32 -11.61  13.3    7.03   4.46]
 [ 21.33  -5.38   1.22 -27.28   9.66   4.85]
 [  0.04  23.35   0.25  -8.53 -17.54   8.81]
 [ 18.35  10.66  13.71  -6.33 -25.3    2.81]
 [ 11.3   11.05   9.05   2.22   5.79   1.02]
 [ 13.79   8.55   3.46   5.23   1.02   4.54]]
##### Matrix #####
 [[ -3.86   5.32 -11.61  13.3    7.03]
 [ 21.33  -5.38   1.22 -27.28   9.66]
 [  0.04  23.35   0.25  -8.53 -17.54]
 [ 18.35  10.66  13.71  -6.33 -25.3 ]
 [ 11.3   11.05   9.05   2.22   5.79]]
#### Removing 3 row And 4 column ####
 [[ -3.86   5.32 -11.61   7.03]
 [ 21.33  -5.38   1.22   9.66]
 [ 18.35  10.66  13.71 -25.3 ]
 [ 11.3   11.05   9.05   5.79]]
Maximum value in absolute matrix ==>  27.28
Minimum value in absolute matrix ==>  0.04
Sum of all elements in absolute matrix ==>  269.46000000000004
```