Lab test 3

# Program 1

Write a Python program using NumPy to generate a 2x4 dimensional array with random values. From the generated array of values do the following operations.

* Find the square root of each value and make a new array from it.
* Concatenate original array and square root array.
* Count and display the number of even and odd values from a concatenated array.

## Code

import numpy as np

A = np.random.rand(2, 4)

*# The range of random values is 0-1.*

*# To increase the magnitude and make the elements integers...*

A = (A \* 10).astype(int)

print("\nNUMPY ARRAY HANDLING")

print("========================")

print("Random array:\nA =")

print(A)

print("========================")

*# SQUARE ROOT ARRAY*

B = np.sqrt(A)

print("Square roots of elements A:\nB =")

print(B)

print("========================")

*# CONCATENTATION OF ARRAYS*

C = np.concatenate((A, B), axis = None)

*# axis = None ensures that the resultant is a 1D array*

print("Concatenation of A and B:\nC = [")

for i in C: print(i, end = ",\n")

print("]")

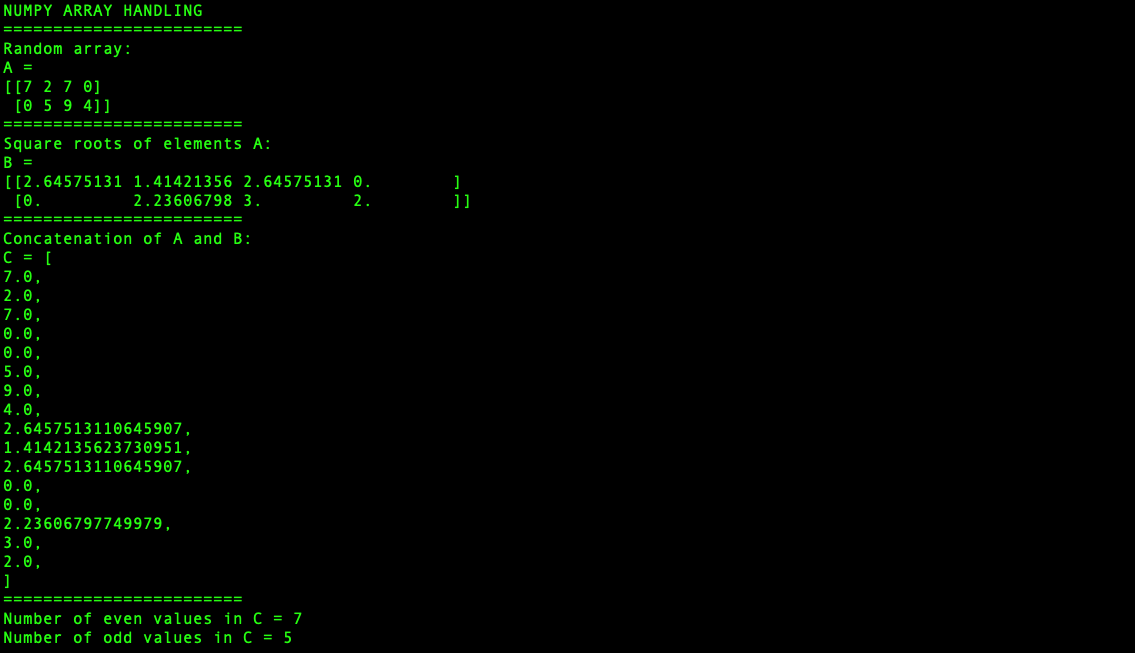
print("========================")

print("Number of even values in C =", np.sum(C % 2 == 0))

print("Number of odd values in C =", np.sum(C % 2 == 1))

print("\n")

## Output



# Program 2

Fetch an array of employee details from **employee.csv**(attachment) such as emp id, basic salary, age, and experience details. Perform the following tasks using NumPy.

* Find and display the youngest employee.
* Find and display the most experienced employee.
* Find the average salary of the employees whose age is between 30 to 40.

## Code

import numpy as np

import pandas as pd

*# Converting dataframe to a 2D numpy array...*

myDataFrame = pd.read\_csv('employee.csv')

data = myDataFrame.to\_numpy()

print("\nEMPLOYEE DATA HANDLING RESULTS")

print("====================================")

*# YOUNGEST EMPLOYEE*

ages = data[:, 2]

minimumAge = min(ages)

isMinimum = (ages == minimumAge)

*# isMinimum is an array of Boolean elements*

*# (corresponding to whether a the age is the minimum age)*

*# Obtaining the record of the employee(s) with the minimum age...*

employeesWithMinimumAge = data[isMinimum]

print("The employee(s) with the minimum age have the following details:")

print("-- Columns:", myDataFrame.columns.values, "--")

print(employeesWithMinimumAge)

print("====================================")

*# MOST EXPERIENCED EMPLOYEE*

xp = data[:, 3]

maximumExperience = max(xp)

isMaximum = (xp == maximumExperience)

*# isMaximum is an array of Boolean elements*

*# (corresponding to whether a the experience is the maximum)*

*# Obtaining the record of the employee(s) with the minimum age...*

employeesWithMaximumXp = data[isMaximum]

print("The employee(s) with the maximum experience have the following details:")

print("-- Columns:", myDataFrame.columns.values, "--")

print(employeesWithMaximumXp)

print("====================================")

*# AVERAGE SALARY OF EMPLOYEES BETWEEN AGES 30 AND 40*

ages = data[:, 2]

isWithinGivenRange = (ages > 30) & (ages < 40)

*# isWithinGivenRange is an array of Boolean elements*

*# (corresponding to whether a the age is bewteen 30 and 40)*

salaries = data[:, 1]

salariesForAgesWithinGivenRange = salaries[isWithinGivenRange]

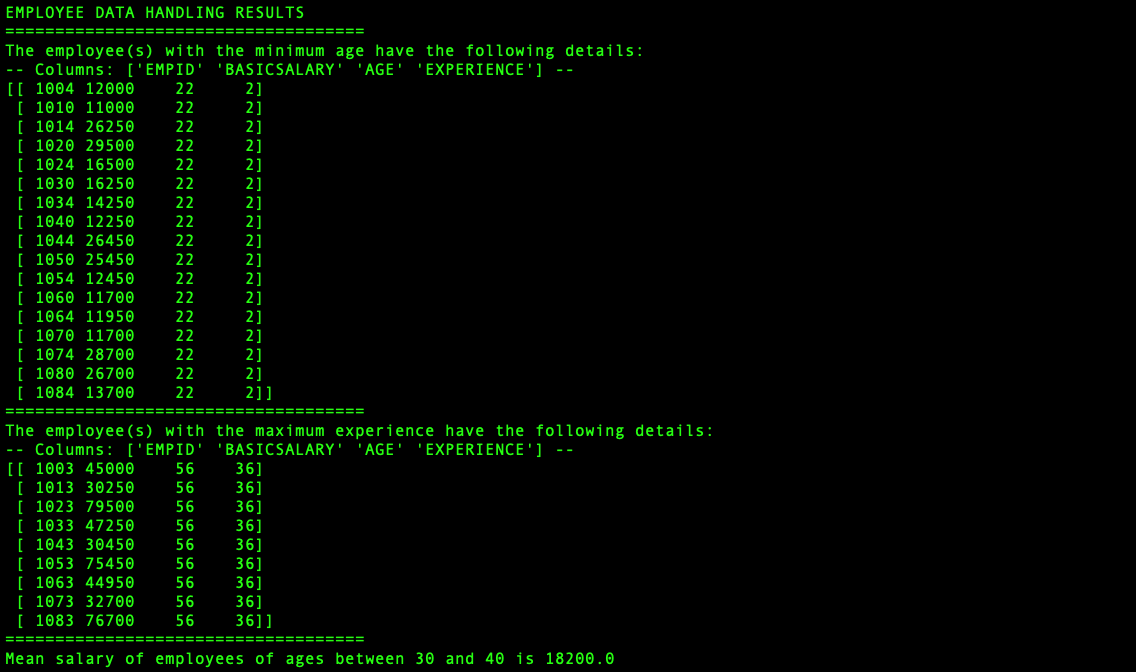
*# The average of the obtained values...*

print("Mean salary of employees of ages between 30 and 40 is", end = " ")

print(np.mean(salariesForAgesWithinGivenRange))

print("\n")

## Output



## Inferences

Converting a Pandas data frame into a NumPy array loses the details, but allows you to treat each row as an array, and the whole data frame as a two-dimensional array. This makes computations and comparisons much faster and more compactly performed, especially more complex comparisons, since array computation in NumPy is many magnitudes faster than looping in Python.