Experiment No: 7

7. Write a NumPy program to find the number of elements of an array, length of one array element in bytes and total bytes consumed by the elements.

Aim: To write a NumPy program to find the number of elements of an array, length of one array element in bytes and total bytes consumed by the elements.

Description:

Numpy size() : In Python, numpy.size() function count the number of elements along a given axis.
Syntax: numpy.size(arr, axis=None)
Parameters:
arr: [array_like] Input data.
axis: [int, optional] Axis(x,y,z) along which the elements(rows or columns) are counted. By default, give the total number of elements in a array
Returns: [int] Return the number of elements along a given axis.

numpy.ndarray.itemsize() : numpy.ndarray.itemsize() function return the length of one array element in bytes.
Syntax : numpy.ndarray.itemsize(arr)
Parameters :
arr : [array_like] Input array.
Return : [int] The length of one array element in bytes.

numpy.ndarray.nbytes() : numpy.ndarray.nbytes() function return total bytes consumed by the elements of the array.
Syntax : numpy.ndarray.nbytes(arr)
Parameters :
arr : [array_like] Input array.
Return : [int] Total bytes consumed by the elements of the array.

Program:

```
import numpy as np

array = np.array([1,2,3])

print("Size of the array: ", array.size)

print("Length of one array element in bytes: ", array.itemsize)

print("Total bytes consumed by the elements of the array: ", array.nbytes)
```

Output:

```
Size of the array:  3

Length of one array element in bytes:  8

Total bytes consumed by the elements of the array:  24
```

8. Write a Pandas program to create

a)Datetime object for Jan 15 2012.

b)Specific date and time of 9:20 pm.

c)Local date and time.

d)A date without time.

e)Current date.

f)Time from a datetime.

g)Current local time

Aim: To write a pandas program to create the above mentioned conditions.

Description:

Python datetime module :

In Python, date and time are not a data type of their own, but a module named datetime can be imported to work with the date as well as time. Python Datetime module comes built into Python, so there is no need to install it externally. Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.
The DateTime module is categorized into 6 main classes –

- **date** – An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.
- **time** – An idealized time, independent of any particular day, assuming that every day has exactly 24*60*60 seconds. Its attributes are hour, minute, second, microsecond, and tzinfo.
- **datetime** – Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.
- **timedelta** – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.
- **tzinfo** – It provides time zone information objects.
- **timezone** – A class that implements the tzinfo abstract base class as a fixed offset from the UTC (New in version 3.2).

Date class

The date class is used to instantiate date objects in Python. When an object of this class is instantiated, it represents a date in the format YYYY-MM-DD. Constructor of this class needs three mandatory arguments year, month and date.

Constructor syntax:
class datetime.date(year, month, day)

The arguments must be in the following range –

- MINYEAR <= year <= MAXYEAR
- 1 <= month <= 12
- 1 <= day <= number of days in the given month and year

Note – If the argument is not an integer it will raise a TypeError and if it is outside the range a ValueError will be raised.

Program:

```
import datetime

from datetime import datetime

print("Datetime object for Jan 15 2012")

print(datetime(2012,1,15))

print("Specific date and time of 9:20 pm")

print(datetime(2010,2,16,21,20))

print("Local date and time")

print(datetime.now())

print("A date without time")

print(datetime.date(datetime(2020,5,4)))

print("Current date")

print(datetime.now().date())

print("Time from a datetime")

print(datetime.time(datetime(2020,5,4)))

print("Current local time")

print(datetime.now().time())
```

Output:

```
Datetime object for Jan 15 2012

2012-01-15 00:00:00

Specific date and time of 9:20 pm

2010-02-16 21:20:00

Local date and time

2022-04-13 08:32:03.035008

A date without time

2020-05-04

Current date

2022-04-13

Time from a datetime

00:00:00

Current local time

08:32:03.038477
```