Roll No:- _____

Sonopant Dandekar Shikshan Mandali's

Sonopant Dandekar Arts,V.S.Apte Commerce,

M.H.Mehta Science College

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

Certified That Mr./Miss._____

of_____has satisfactorily completed a course of necessary experiment in _____under my supervision in the SY.BSC Computer Science in the Year 2025 – 2026.

Head of Department **Subject Teacher**

Date: / / 2025

SR No.	Aim of Practical	Practical Date	Submission Date	Remarks
1.	A]Write a python program to print your profile B]. Write a python program to print addition of two numbers. C]. Write a python program to print square root of number. D]. Write a python program to calculate area of triangle.			
2.	A]Write a python code to swap two variables. B]Write a python code to create nested tuples. C]Write a python code to sort the nested tuple using sorted () function. D]Write a python code to copy or clone list. E] Write a python code to check immutability property of python tuples.			
3.	A] Write a python code for creating variable and storing the text that we want to search. B] Write a python code to retrieve data from HTML file. C] Write a python code to print current date in different format.			
4.	A] Write a python code to print current date in different formats. B] Write a python code to develop calendar module. C] Write a python code to compare two dates. D] Write a python code to convert time stamp to date stamp			
5.	A] Write a python code to create NumPy array B] Write a python code to demonstrate basic operation on array			
6.	Write the python code to create array with an element and to perform various method Joining Array, Sorting Array, Spliting Array, Searching Array			
7.	A] Write a python code on basic operation on single Array. B] Write a python code to create array with 10 elements and slice element from 1 to 5 elements. C] Write a python code to sort an array alphabetically.			
8.	Write a python code to demonstrate pandas libraries and create dataframe.			
9.	Write a python code that shows statistical information for given dataset.			
10.	Write a python code to demonstrate filter pandas series with Boolean Arrays			

Date:_____

Practical No 1

Aim: A]. Write a python program to print your profile.

B]. Write a python program to print addition of two numbers.

C]. Write a python program to print square root of number.

D]. Write a python program to calculate area of triangle.

A. Write a python program to print your profile.

Input:

```
from tkinter import*
master=Tk()
Label(master,text="first Name").grid(row=0)
Label(master,text="Last Name").grid(row=1)
Label(master,text="Age").grid(row=2)
Label(master,text="Class").grid(row=3)
master.configure(bg="pink")
e1=Entry(master)
e2=Entry(master)
e3=Entry(master)
e4=Entry(master)
e1.grid(row=0,column=1)
e2.grid(row=1,column=1)
e3.grid(row=2,column=1)
e4.grid(row=3,column=1)
mainloop()
```

Output:

OR

Input:

```
def personal_details():
    name, age="Reema", 19
    address="Banglore,karnataka,india."
    roll_no="002"
```

```
    Class="SYCS"
print("\nName:{}\nAge:{}\nAddress:{}\nRoll_no:{}\nClass:{}".format(name,age,address,roll_no,Class))
personal_details()
```

Output:

```
Name:Reema
Age:19
Address:Banglore,karnataka,india.
Roll_no:002
Class:SYCS
```

B]. Write a python program to print addition of two numbers.

Input:

```
num1=int(input("Enter the First number."))
num2=int(input("Enter the Second number."))
add=int(num1)+ int(num2)
print(add)
```

Output:

```
========================================== RES
Enter the First number.45
Enter the Second number.78
123
```

OR By using Function

Input:

```
def add():
    num1=int(input("Enter the First number."))
    num2=int(input("Enter the Second number."))
    result=int(num1)+ int(num2)
    print(result)
add()
```

Output:

```
Enter the First number.56
Enter the Second number.78
134
```

C]. Write a python program to print square root of number.

Input:

```
# Import math Library
```

```
import math
# Return the square root of different numbers
print (math.sqrt(9))
print (math.sqrt(25))
print (math.sqrt(16))
```

Output:

```
3.0
5.0
4.0
```

OR By user input

Input:

```
import math
num=int(input("Enter a number"))
Sqrt=math.sqrt(num)
print(f"square root of {num} is {Sqrt}")
```

Output:

```
Enter a number4
square root of 4 is 2.0
```

D. Write a python program to calculate area of triangle.

Input:

```
def area_of_traingle():
    base=int(input("Enter the value of base."))
    height=int(input("Enter the value of height."))
    area=0.5* float(base)* float(height)
    print(f"the area of Traingle is :{area}")
area_of_traingle()
```

Output:

```
Enter the value of base.34
Enter the value of height.23
the area of Traingle is :391.0
```

\

Date:_____

Practical No 2

Aim: A]Write a python code to swap two variables.

B]Write a python code to create nested tuples.
C]Write a python code to sort the nested tuple using sorted () function.
D]Write a python code to copy or clone list.
E] Write a python code to check immutability property of python tuples.

A] Write a python code to swap two variables.

Input:

1.
```
a=18
b=24
#Swapping
temp=a
a=b
b=temp
print("a=",a)
print("b=",b)
```
```
a= 24
b= 18
```

2.

```
a=float(input("Enter first number"))
b=float(input("Enter second number"))
temp=a
a=b
b=temp
print("a=",a)
print("b=",b)
```
```
Enter first number18.5
Enter second number7.9
a= 7.9
b= 18.5
```

B] Write a python code to create nested tuples.

Input:

```
tuple1=(1,2,3)

tuple2=('a','b','c')

nested_tuple=(tuple1,tuple2)

print("tuple1:",tuple1)

print("tuple2:",tuple2)
```

```
print("Nested tuple:",nested_tuple)
```

```
tuple1: (1, 2, 3)
tuple2: ('a', 'b', 'c')
Nested tuple: ((1, 2, 3), ('a', 'b', 'c'))
```

C] Write a python code to sort the nested tuple using sorted () function.

1. Simple

Input:

```
#original nested tuple
nested_tuple=((3,'banana'),(1,'apple'),(2,'cherry'))
# sort the nested tuple by the first element of each inner tuple
sorted_nested=tuple(sorted(nested_tuple))
#print result
print("Original:",nested_tuple)
print("Sorted:",sorted_nested)
```

```
= RESTART: C:/Users/student/Desktop/65014py/sort.py
Original: ((3, 'banana'), (1, 'apple'), (2, 'cherry'))
Sorted: ((1, 'apple'), (2, 'cherry'), (3, 'banana'))
```

2. Sort by First Element (By default)

Input:

```
data=((4,'Dog'),(2,'Cat'),(3,'Bear'),(1,'Ant'))
sorted_nested=tuple(sorted(data))
print("Original:",data)
print("Sorted:",sorted_nested)
```

```
========== RESTART: C:/Users/student/Desktop/65014py/sort.py
Original: ((4, 'Dog'), (2, 'Cat'), (3, 'Bear'), (1, 'Ant'))
Sorted: ((1, 'Ant'), (2, 'Cat'), (3, 'Bear'), (4, 'Dog'))
```

3. Sort by Second Element (By Alphabetically)

Input:

```
data=((4,'Dog'),(2,'Cat'),(3,'Bear'),(1,'Ant'))
sorted_second=tuple(sorted(data,key=lambda x:x[1]))
print("Original:",data)
print("Sorted by second Element:",sorted_second)
```

```
========== RESTART: C:/Users/student/Desktop/65014py/sort.py =========
Original: ((4, 'Dog'), (2, 'Cat'), (3, 'Bear'), (1, 'Ant'))
Sorted by second Element: ((1, 'Ant'), (3, 'Bear'), (2, 'Cat'), (4, 'Dog'))
```

4. sort in Reverse order(Descending by first Element)

Input:

```
data=((4,'dog'),(2,'Cat'),(3,'Bear'),(1,'Ant'))
sort_desc=tuple(sorted(data,reverse=True))
```

```
print('Original:',data)

print("Sorted descending by first element:",sort_desc)
```

```
========== RESTART: C:/Users/student/Desktop/65014py/sort.py ==========
Original: ((4, 'dog'), (2, 'Cat'), (3, 'Bear'), (1, 'Ant'))
Sorted descending by first element: ((4, 'dog'), (3, 'Bear'), (2, 'Cat'), (1, 'Ant'))
```

D] Write a python code to copy or clone list.

1.

Input:

```
a=[1,2,3,4,5]
b=a.copy()
print("Original:",a)
print("copied:",b)
```

```
Original: [1, 2, 3, 4, 5]
copied: [1, 2, 3, 4, 5]
```

2.

```
a=["Apple","banana"]
b=a.copy()
print("Original:",a)
print("copied:",b)
```

```
Original: ['Apple', 'banana']
copied: ['Apple', 'banana']
```

3.

Input:

```
thislist=["apple","banana","cherry"]
mylist=list(thislist)
print(mylist)
```

```
['apple', 'banana', 'cherry']
```

E] Write a python code to check immutability property of python tuples.

```
data=(11,12,13,14)

print("Original:",data)

data[2]=15

print("Modified tuple is:",data)
```

Output:

```
Original: (11, 12, 13, 14)
Traceback (most recent call last):
  File "C:/Users/student/AppData/Local/Programs/Python/Python312/64032/d.py", li
ne 3, in <module>
    data[2]=15
TypeError: 'tuple' object does not support item assignment
```

Date: _____

Practical No 3

Aim: A] Write a python code for creating variable and storing the text that we want to search.

B] Write a python code to retrieve data from HTML file.

C] Write a python code to print current date in different format.

A] Write a python code for creating variable and storing the text that we want to search.

Input:

```
# create a varible to store value
search_text="Machine Learning."
print(search_text)
```

Output:

```
Machine Learning.
```

B] Write a python code to retrieve data from HTML file.
Note: First install library Beautifulsoup.
Cmd: pip install beautifulsoup4 requests.

```
>pip install beautifulsoup4 requests
```

Then make two files 1. Python code file 2. Html code file.

1] First Example: Extract the text from all tags.

1. python file

Input:

```
from bs4 import BeautifulSoup
#Read HTML  content from a file
with open("D:/65014py/example1.html",'r',encoding='utf-8')as file:
    html_cont=file.read()
    #parse the Html content
soup=BeautifulSoup(html_cont,'html.parser')
#Extract text from all tags
all_tags=soup.find_all()
for tag in all_tags:
    print(tag.get_text())
```

2. html code

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Extracting Html text</title>
```

```
</head>
<body>
<h1> Welcome to SYBSC-CS</h1>
<div class="content">
<p> Hello Everyone!!!</p>
</div>
</body>
</html>
```

Output:

```
Extracting Html text

Extracting Html text
 Welcome to SYBSC-CS
 Hello Everyone!!!

 Welcome to SYBSC-CS
 Hello Everyone!!!
 Hello Everyone!!!
```

2] Second Example: Extract the text from nested tags.

1. python code:

Input:

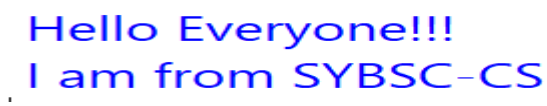
```
from bs4 import BeautifulSoup
#Read HTML  content from a file
with open("D:/65014py/example1.html",'r',encoding='utf-8')as file:
    html_cont=file.read()
    #parse the Html content
soup=BeautifulSoup(html_cont,'html.parser')
#Extract the text from nested tag
divs=soup.find_all('div',class_='content')
for div in divs:
    paragraphs=div.find_all('p')
    for p in paragraphs:
        print(p.get_text())
```

2. html code:

```
<!doctype html>
```

```
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Extracting Html text</title>
</head>
<body>
<h1> Welcome to SYBSC-CS</h1>
<div class="content">
<p> Hello Everyone!!!</p>
<p> I am from SYBSC-CS</p>
</div>
</body>
</html>
```

Output:



3] Third Example: Extracting using CSS selector.

1. python code:

Input:

```
from bs4 import BeautifulSoup
#Read HTML  content from a file
with open("D:/65014py/example2.html",'r',encoding='utf-8')as file:
    html_cont=file.read()
    #parse the Html content
soup=BeautifulSoup(html_cont,'html.parser')
texts=soup.select('div.content>p')# select all<p> tags inside <div class="content">
for text in texts:
    print(text.get_text())
```

2. html code:

Input:

```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
<h1> Welcome to SYBSC-CS</h1>
```

```
<div class="content">
<p> Hello Everyone!!!</p>
<p> This is the Example of CSS Selector.</p>
</div>
</body>
</html>
```

Output:

> Hello Everyone!!!
> This is the Example of CSS Selector.

C] Write a python code to print current date in different format.

Input:

```
from datetime import datetime
#get current date and time
now=datetime.now()
#print current date in various format
print("Default format:",now)
print("Date only(YYYY-MM-DD):",now.strftime("%Y-%m-%d"))
print("Date(DD/MM/YYYY):",now.strftime("%d/%m/%Y"))
print("Date(MM-DD-YYYY):",now.strftime("%m-%d-%Y"))
print("Full Month Name,Day Year:",now.strftime("%B %d;%Y"))
print("Day-Month Abbreviation-Year:",now.strftime("%d-%b-%Y"))
print("Weekday,Month Day,Year:",now.strftime("%A,%B%d,%Y"))
print("ISO Format:",now.isoformat())
```

Output:

```
Default format: 2025-08-11 11:38:46.850893
Date only(YYYY-MM-DD): 2025-08-11
Date(DD/MM/YYYY): 11/08/2025
Date(MM-DD-YYYY): 08-11-2025
Full Month Name,Day Year: August 11;2025
Day-Month Abbreviation-Year: 11-Aug-2025
Weekday,Month Day,Year: Monday,August11,2025
ISO Format: 2025-08-11T11:38:46.850893
```

Date:_____

Practical No 4

Aim: A] Write a python code to print current date in different formats.

B] Write a python code to develop calendar module.

C] Write a python code to compare two dates.

D] Write a python code to convert time stamp to date stamp

A] Write Current date in different format

1] Write a pyhon code to print current date

Input:

```
from datetime import date
today=date.today()
print(f"Today's date:{today}")
```

Output:

```
Today's date:2025-08-16
```

2] print current time in milliseconds.

Input:

```
import time
milliseconds=int(round(time.time()*1000))
print(milliseconds)
```

Output:

```
1755320102536
```

3] Convert following datetime into a string

Input:

```
from datetime import datetime
given_date=datetime(2020,2,25)
string_date=given_date.strftime("%Y-%m-%d%H:%M:%S")
print(string_date)
```

Output:

```
2020-02-2500:00:00
```

4] display current date and time in different format.

Input:

```
import datetime
x=datetime.datetime.now()
print(f"only short form day of Week:, {x.strftime("%a")}")
print(f"today day is :,{x.strftime("%A")}")
print(f"no of day of week :,{x.strftime("%w")}")
print(f"Date  :,{x.strftime("%d")}")
print(f"In MM_DD_YY format:,{x.strftime("%D")}")
print(f"Month:,{x.strftime("%m")}")
print(f"minute as a zero-padded decimal number.:,{x.strftime("%M")}")
print(f"Only Month:,{x.strftime("%B")}")
print(f" Week number of the Year:,{x.strftime("%U")}")
print(f"Only short form of yera:,{x.strftime("%y")}")
print(f"YYYY format:,{x.strftime("%Y")}")
print({x.strftime("%S")})
print({x.strftime("%p")})
```

Output:

```
only short form day of Week:, Sat
today day is :,Saturday
no of day of week :,6
Date  :,16
In MM_DD_YY format:,08/16/25
Month:,08
minute as a zero-padded decimal number.:,51
Only Month:,August
 Week number of the Year:,32
Only short form of yera:,25
YYYY format:,2025
{'32'}
{'AM'}
```

B] Write a python code to develop calendar module.

1] display simple calendar

Input:

```
import calendar
print(calendar.calendar(2025))
```

Output:

```
                                  2025

      January                   February                  March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
        1  2  3  4  5                1  2                      1  2
 6  7  8  9 10 11 12       3  4  5  6  7  8  9       3  4  5  6  7  8  9
13 14 15 16 17 18 19      10 11 12 13 14 15 16      10 11 12 13 14 15 16
20 21 22 23 24 25 26      17 18 19 20 21 22 23      17 18 19 20 21 22 23
27 28 29 30 31            24 25 26 27 28            24 25 26 27 28 29 30
                                                    31

      April                     May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
     1  2  3  4  5  6                1  2  3  4                         1
 7  8  9 10 11 12 13       5  6  7  8  9 10 11       2  3  4  5  6  7  8
14 15 16 17 18 19 20      12 13 14 15 16 17 18       9 10 11 12 13 14 15
21 22 23 24 25 26 27      19 20 21 22 23 24 25      16 17 18 19 20 21 22
28 29 30                  26 27 28 29 30 31         23 24 25 26 27 28 29
                                                    30

      July                    August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
     1  2  3  4  5  6                   1  2  3       1  2  3  4  5  6  7
 7  8  9 10 11 12 13       4  5  6  7  8  9 10       8  9 10 11 12 13 14
14 15 16 17 18 19 20      11 12 13 14 15 16 17      15 16 17 18 19 20 21
21 22 23 24 25 26 27      18 19 20 21 22 23 24      22 23 24 25 26 27 28
28 29 30 31               25 26 27 28 29 30 31      29 30

      October                 November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
        1  2  3  4  5                1  2       1  2  3  4  5  6  7
 6  7  8  9 10 11 12       3  4  5  6  7  8  9       8  9 10 11 12 13 14
13 14 15 16 17 18 19      10 11 12 13 14 15 16      15 16 17 18 19 20 21
20 21 22 23 24 25 26      17 18 19 20 21 22 23      22 23 24 25 26 27 28
27 28 29 30 31            24 25 26 27 28 29 30      29 30 31
```

2] display calendar of full year

Input:

```
import calendar
year=int(input("Enter year:"))
print(calemdar.calendar(year))
```

Output:

```
                                  1997

      January                   February                  March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
        1  2  3  4  5                1  2                      1  2
 6  7  8  9 10 11 12       3  4  5  6  7  8  9       3  4  5  6  7  8  9
13 14 15 16 17 18 19      10 11 12 13 14 15 16      10 11 12 13 14 15 16
20 21 22 23 24 25 26      17 18 19 20 21 22 23      17 18 19 20 21 22 23
27 28 29 30 31            24 25 26 27 28            24 25 26 27 28 29 30
                                                    31

      April                     May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
     1  2  3  4  5  6                1  2  3  4                         1
 7  8  9 10 11 12 13       5  6  7  8  9 10 11       2  3  4  5  6  7  8
14 15 16 17 18 19 20      12 13 14 15 16 17 18       9 10 11 12 13 14 15
21 22 23 24 25 26 27      19 20 21 22 23 24 25      16 17 18 19 20 21 22
28 29 30                  26 27 28 29 30 31         23 24 25 26 27 28 29
                                                    30

      July                    August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
     1  2  3  4  5  6                   1  2  3       1  2  3  4  5  6  7
 7  8  9 10 11 12 13       4  5  6  7  8  9 10       8  9 10 11 12 13 14
14 15 16 17 18 19 20      11 12 13 14 15 16 17      15 16 17 18 19 20 21
21 22 23 24 25 26 27      18 19 20 21 22 23 24      22 23 24 25 26 27 28
28 29 30 31               25 26 27 28 29 30 31      29 30

      October                 November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
        1  2  3  4  5                1  2       1  2  3  4  5  6  7
 6  7  8  9 10 11 12       3  4  5  6  7  8  9       8  9 10 11 12 13 14
13 14 15 16 17 18 19      10 11 12 13 14 15 16      15 16 17 18 19 20 21
20 21 22 23 24 25 26      17 18 19 20 21 22 23      22 23 24 25 26 27 28
27 28 29 30 31            24 25 26 27 28 29 30      29 30 31
```

3] display a month of year

Input:

```
import calendar
year=int(input("Enter year:"))
month=int(input("Enter month(1-12:"))
print(calendar.month(year,month))
```

Output:

```
----------------------------
Enter year:2006
Enter month(1-12:12
    December 2006
Mo Tu We Th Fr Sa Su
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

4] display that the year is leap or not

Input:

```
import calendar
year=int(input("Enter year:"))
if calendar.isleap(year):
   print(f"{year} is a leap year ")
else:
   print(f"{year} is NOT a leap year")
```

Output:

```
==========================================
Enter year:1999
1999 is NOT a leap year

==========================================
Enter year:2024
2024 is a leap year
```

C] Write a python code to compare two dates.

Input:

```
from datetime import datetime
#2025-08-16
```

```
date_1=datetime(2025,8,16).date()
# 2020-09-17
date_2=datetime(2020,9,17).date()
delta=None
if date_1>date_2:
    print("date_1 is greater")
    delta=date_1-date_2
else:
    print("date_2 is greater")
    delta=date_2-date_1
print("Difference is",delta.days,"days")
```

Output:

```
==========
date_1 is greater
Difference is 1794 days
```

D] Write a python code to convert time stamp to date stamp

Input:

```
from datetime import datetime
timestamp_string="2025-08-16 15:30:45"
format_string="%Y-%m-%d %H:%M:%S"
datetime_object=datetime.strptime(timestamp_string,format_string)
print(datetime_object)
```

Output:

```
==========
2025-08-16 15:30:45
```

Date: _____

Practical No 5

Aim: A] Write a python code to create NumPy array

B] Write a python code to demonstrate basic operation on array

---

Note: first install numpy

```
C:\Users\student>pip install numpy
```

A] Write a python code to create NumPy array

1. first Example: to display array.

Input:

```
import numpy
arr=numpy.array([2,3,4])
print(arr)
```

Output:

```
[2 3 4]
```

2. Second Example: to display array by using as keyword.

Input:

```
import numpy as np
arr=np.array([1,8,2,4])
print(arr)
```

Output:

```
[1 8 2 4]
```

3. Third Example : to display array and its type

Input:

```
import numpy as np

arr=np.array([14,56,79,39])

print(arr)

print(type(arr))
```

Output:

```
[14 56 79 39]
<class 'numpy.ndarray'>
```

4. fourth Example: To display the array according to index.

Input:

```
import numpy as np
arr=np.array([18,24,18,12,10])
print(arr)
print(f" Element at index o is:",arr[0])
print(f" Element at index 1 is:",arr[1])
print(f" Element at index 2 is:",arr[2])
print(f" Element at index 3 is:",arr[3])
print(f" Element at index 4 is:",arr[4])
```

Output:

```
[18 24 18 12 10]
 Element at index o is: 18
 Element at index 1 is: 24
 Element at index 2 is: 18
 Element at index 3 is: 12
 Element at index 4 is: 10
```

5. Fifth Example: Array Slicing.

Input:

```
import numpy as np
arr=np.array([24,18,10,12,78,90])
print(f"Original array:{arr}")
print(f"Print Element at 5Th position: {arr[5:]}")
print(f"print First five elements: {arr[:5]}")
```

Output:

```
Original array:[24 18 10 12 78 90]
Print Element at 5Th position: [90]
print First five elements: [24 18 10 12 78]
```

6. creating a simple and nested array

Input:

```
# creating a rank 1 Array
import numpy as np
arr=np.array([24,18,10,12,78,90])
print(arr)
# creating a rank 2 Array
arr1=np.array([[1,2,3],[8,9,7]])
print(arr1)
```

Output:

```
[24 18 10 12 78 90]
[[1 2 3]
 [8 9 7]]
```

7. creating an array from tuple

Input:

```
import numpy as np
arr=np.array({1,3,4,5,3,2,2,})
print(f"Display Tuple",arr)
```

Output:

```
Display Tuple {1, 2, 3, 4, 5}
```

B] Write a python code to demonstrate basic operation on array

1. Creating Arrays.

Input:

```
import numpy as np
array1=np.array([1,4,5,7])#1D
array2=np.array([[2,4],[6,9]])#2D
print("Array 1:",array1)
print("Array2:",array2)
```

Output:

```
Array 1: [1 4 5 7]
Array2: [[2 4]
 [6 9]]
```

2. Basic Arithmetic Operations

Input:

```
import numpy as np

array1=np.array([23,58,90,78])

array2=np.array([34,67,86,23])

print("Array 1:",array1)

print("Array2:",array2)

print("Addition:", array1+array2)

print("Subtraction:", array1-array2)

print("Multiplication:", array1*array2)

print("Division:", array1/array2)
```

Output:

```
Array 1: [23 58 90 78]
Array2: [34 67 86 23]
Addition: [ 57 125 176 101]
Subtraction: [-11  -9   4  55]
Multiplication: [ 782 3886 7740 1794]
Division: [0.67647059 0.86567164 1.04651163 3.39130435]
```

3. Mathematical Function

Input:

```
import numpy as np
array1=np.array([4,9,16,25])
print("Square Root of Array 1:",np.sqrt(array1))
print("Exponential:",np.exp(array1))
print("Sine of array 1:",np.sin(array1))
```

Output:

```
Square Root of Array 1: [2. 3. 4. 5.]
Exponential: [5.45981500e+01 8.10308393e+03 8.88611052e+06 7.20048993e+10]
Sine of array 1: [-0.7568025   0.41211849 -0.28790332 -0.13235175]
```

4. Aggregation Function

Input:

```
import numpy as np
array1=np.array([4,9,16,25])
print("Original array",array1)
print("Sum of array:",np.sum(array1))
print("Mean of array:",np.mean(array1))
print("Maximum No of array:",np.max(array1))
print("Minimum No of array:",np.min(array1))
```

Output:

```
Original array [ 4  9 16 25]
Sum of array: 54
Mean of array: 13.5
Maximum No of array: 25
Minimum No of array: 4
```

Date: _____

Practical No 6

Aim: Write the python code to create array with an element and to perform various method

A] Joining Array
B] Sorting Array
C] Spliting Array
D] Searching Array

A] Joining Array: using concatenate method.

Input:

```
import numpy as np
arr1=np.array([[2,4],[1,8]])
arr2=np.array([[1,0],[1,2]])
arr=np.concatenate((arr1,arr2),axis=1)
print(f"First Array:",arr1)
print(f"Second Array:",arr2)
print(f"After Joining:",arr)
```

Output:

```
First Array: [[2 4]
 [1 8]]
Second Array: [[1 0]
 [1 2]]
After Joining: [[2 4 1 0]
 [1 8 1 2]]
```

B] Sorting Array:

1. In Ascending Order

Input:

```
import numpy as np
arr=np.array([3,6,1,9,4])
new_arr=np.sort(arr)
print("Original array:",arr)
print("After Sorting in Ascending order:",new_arr)
```

Output:

```
Original array: [3 6 1 9 4]
After Sorting in Ascending order: [1 3 4 6 9]
```

2. In Descending Order

Input:

```
import numpy as np
arr = np.array([3, 6, 1, 9, 4])
new_arr = np.sort(arr)[::-1]  # Sort in ascending and then reverse for descending
print("Original array:", arr)
print("After Sorting in Descending order:", new_arr)
```

Output:

```
-------
Original array: [3 6 1 9 4]
After Sorting in Descending order: [9 6 4 3 1]
```

C] Spliting Array: Using array_split function.

Input:

```
import numpy as np
arr=np.array([3,6,1,9,4])
new_arr=np.array_split(arr,3)
print(new_arr)
```

Output:

```
[array([3, 6]), array([1, 9]), array([4])]
```

D] Searching Array: Using Where method.

Input:

```
import numpy as np
arr=np.array([2,9,5,3,4,8])
x=np.where(arr==5)
print("Original array:",arr)
print(f"On which position is No 5:",x)
```

Output:

```
=======
Original array: [2 9 5 3 4 8]
On which position is No 5: (array([2]),)
```

Date:_____

Practical No 7

Aim: A] Write a python code on basic operation on single Array.

B] Write a python code to create array with 10 elements and slice element from 1 to 5 elements.

C] Write a python code to sort an array alphabetically.

A] Write a python code on basic operation on single Array.

1] basic operation on single array.

Input:

```
#creation of array
my_list=[10,20,30,40,50]
#Accesing an array
print("first element :",my_list[0])
print("Third element :",my_list[3])
print("Last element :",my_list[-1])
#modifying an element
my_list[2]=18
print("Array after modificcation:",my_list)
#adding an element
my_list.append(24)
print("Array after adding:",my_list)
#removing an element
my_list.remove(40)
print("Array after removing:",my_list)
#sum of element
arr=sum(my_list)
print("Array after sum of array:",arr)
#length of an array
length=len(my_list)
print("length of array:",length)
```

```
#insertion
my_list.insert(2,78)
print(my_list)
```

Output:

```
first element : 10
Third element : 40
Last element : 50
Array after modificcation: [10, 20, 18, 40, 50]
Array after adding: [10, 20, 18, 40, 50, 24]
Array after removing: [10, 20, 18, 50, 24]
Array after sum of array: 122
length of array: 5
[10, 20, 78, 18, 50, 24]
```

2] basic operation on single array(using Numpy).

Input:

```
import numpy as np
#creation of array
my_list=np.array([10,20,30,40,50])
#Accesing an array
print("first element :",my_list[0])
print("Third element :",my_list[3])
print("Last element :",my_list[-1])
#modifying an element
my_list[2]=18
print("Array after modificcation:",my_list)
#adding an element
my_list=np.append(my_list,60)
print("Array after adding:",my_list)
#insertion
my_list=np.insert(my_list,2,78)
print(my_list)
#removing an element
my_list=np.delete(my_list,4)
print("Array after removing:",my_list)
```

```
#sum of element
arr=np.sum(my_list)
print("Array after sum of array:",arr)
#length of an array
length=len(my_list)
print("length of array:",length)
```

Output:

```
first element : 10
Third element : 40
Last element : 50
Array after modificcation: [10 20 18 40 50]
Array after adding: [10 20 18 40 50 60]
[10 20 78 18 40 50 60]
Array after removing: [10 20 78 18 50 60]
Array after sum of array: 236
length of array: 6
```

B] Write a python code to create array with 10 elements and slice element from 1 to 5 elements.

Input:

```
my_list=([29,15,10,18,24,14,6,12,7,28])
#slicing the list fom index 1 to 5
sliced_list=my_list[1:6]
print("Original list:",my_list)
print("Sliced list(element from 1 to 5):",sliced_list)
```

Output:

```
Original list: [29, 15, 10, 18, 24, 14, 6, 12, 7, 28]
Sliced list(element from 1 to 5): [15, 10, 18, 24, 14]
```

C] Write a python code to sort an array alphabetically.

Input:

```
my_list=["Dragonfruit","Banana","Apple","Blueberry","Chickoo","Pineapple","Kiwi"]
my_list.sort()
print("Sorted list by alphabetically:",my_list)
```

Output:

```
Sorted list by alphabetically: ['Apple', 'Banana', 'Blueberry', 'Chickoo', 'Dragonfruit', 'Kiwi', 'Pineapple']
```

Date:_____

Practical No 8

Aim: Write a python code to demonstrate pandas libraries and create dataframe.

Install pandas library: pip install pandas

```
C:\Users\student>pip install pandas
```

Example 1: create simple dataframe using pandas

Input:

```
import pandas as pd
data={
   "Student_Name":['abc','pqr','xyz'],
   "Age":[25,30,50],
   "Class":['Fycs','Sycs','Tycs']
   }
df=pd.DataFrame(data)
print(df)
```

Output:

```
  Student_Name  Age Class
0          abc   25  Fycs
1          pqr   30  Sycs
2          xyz   50  Tycs
```

Example 2: Form a list of dictoneries.

Input:

```
import pandas as pd
data = [
   {'Name':"ABC",'Age':25},
   {'Name':"XYZ",'Age':30},
   {'Name':"PQR",'Age':45}
]
df = pd.DataFrame(data)
print(df)
```

Output:

```
  Name  Age
0  ABC   25
1  XYZ   30
2  PQR   45
```

Example 3: form a dictionary of series

Input:

```
import pandas as pd
data={
   'Name':pd.Series(['abc','pqr','xyz']),
   'Age':pd.Series([29,38,48])
   }
df=pd.DataFrame(data)
print(df)
```

Output:

```
  Name  Age
0  abc   29
1  pqr   38
2  xyz   48
```

Example 4: Form a list of lists with column names

Input:

```
import pandas as pd
data=[
   ['ABC',34,'Fycs'],
   ['XYZ',24,'Sycs'],
   ['PQR',18,'Tycs']
   ]
columns=['Name','Age','Class']
df=pd.DataFrame(data,columns)
print(df)
```

Output:

```
       0   1     2
Name   ABC 34  Fycs
Age    XYZ 24  Sycs
Class  PQR 18  Tycs
```

Example 5: Creating as empty DataFrame and Adding data later

Input:

```
import pandas as pd
#create an empty DataFrame with specific column names
df=pd.DataFrame(columns=['Name','Age','City'])
#append  rows One by one
df.loc[0]=['ABC',14,'New York']
df.loc[1]=['BOB',20,'Los Angeles']
df.loc[2]=['XYZ',19,'Chicago']
print(df)
```

Output:

```
==========================================
  Name  Age       City
0  ABC   14   New York
1  BOB   20  Los Angeles
2  XYZ   19    Chicago
```

Date _____

Practical No 9

Aim: Write a python code that shows statistical information for given dataset.

```
C:\Users\student>pip install numpy
```

Example 1:

Input:

```
import numpy as np
import statistics
data=[34,12,34,56,23,12,90]
print("Mean:",statistics.mean(data))
print("Median:",statistics.median(data))
print("Mode:",statistics.mode(data))
print("Standard Deviation:",statistics.stdev(data))
print("Variance:",statistics.variance(data))
print("Minimum:",min(data))
print("Maximum:",max(data))
```

Output:

```
========== RESTART: E:/65014_py/prac
Mean: 37.285714285714285
Median: 34
Mode: 34
Standard Deviation: 27.789172266156026
Variance: 772.2380952380952
Minimum: 12
Maximum: 90
```

Example 2:

Input:

```
import numpy as np
import pandas as pd
from scipy import stats
# sample dataset (You can replace it with your own data)
data=[29,15,10,24,18,6,30,14,21,28,3]
#convert to pandas Series for easy statistics
series=pd.Series(data)
#statistical information
print("====Satistical Summary====")
print(f"Count  : {series.count()} ")
print(f"Mean  : {series.mean():.2f} ")
print(f"Median  : {series.median():.2f} ")
print(f"Mode  : {series.mode().tolist()} ")
print(f"Standard deviation  : {series.std():.2f} ")
print(f"Variance : {series.var():.2f} ")
print(f"Minimum  : {series.min()} ")
print(f"Maximum  : {series.max()} ")
print(f"Skewness  : {series.skew():.2f} ")
print(f"Kurtosis : {series.kurt():.2f} ")
```

Output:

```
========== RESTART: E:/65014_py/prac
====Satistical Summary====
Count  : 11
Mean  : 18.00
Median  : 18.00
Mode  : [3, 6, 10, 14, 15, 18, 21, 24, 28, 29, 30]
Standard deviation  : 9.32
Variance : 86.80
Minimum  : 3
Maximum  : 30
Skewness  : -0.21
Kurtosis : -1.18
```

Date: _____

Practical No 10

Aim: Write a python code to demonstrate filter pandas series with Boolean Arrays.

```
C:\Users\student>pip install pandas
```

Example 1:

Input:

```
import pandas as pd
#create a pandas series
data=pd.Series([10,20,30,40,50])
print("Original series:")
print(data)
#Create a Boolean array(same length as ythe series)
bool_array=[True, False,True,False,True]
#filter the series using the Boolean array
filtered_data=data[bool_array]
print("\n Filtered series(using Boolean array):")
print(filtered_data)
```

Output:

```
_______
Original series:
0    10
1    20
2    30
3    40
4    50
dtype: int64

 Filtered series(using Boolean array):
0    10
2    30
4    50
dtype: int64
```

Example 2: Filtering even numbers

Input:

```
import pandas as pd
```

```
data=pd.Series([22,67,44,68,90])
#filter even numbers using modulo
filtered_data=data[data%2==0]
print("Filtered even numbers")
print(filtered_data)
```

Output:

```
Filtered even numbers
0    22
2    44
3    68
4    90
dtype: int64
```

Example 3: Manual Boolean array(custom logic)

Input:

```
import pandas as pd
scores=pd.Series([65,80,45,90,55])
#manual Boolean array (passed/fsil)
bool_array=[False,True,False,True,False]
#filter passed scores
passed_scores=scores[bool_array]
print("passed scores(manual Boolean array):")
print(passed_scores)
```

Output:

```
passed scores(manual Boolean array):
1    80
3    90
dtype: int64
```

Example 4: Filter pandas Dataframe by Column value

Input:

```
import pandas as pd
data={"Name":['Alice','Bob','Charlie'],
    "Age":[25,34,56],
    "score":[86,90,89]}
```

```
df=pd.DataFrame(data)
#Filter rows where Age is greater than 30
filtered_df=df[df['Age']>30]
print(filtered_df)
```

Output:

```
      Name  Age  score
1      Bob   34     90
2  Charlie   56     89
```

Example 5: Filter a number in specific range

Input:

```
import pandas as pd
data=pd.Series([5,12,18,25,30,40])
#filter numbers between 15 and 35(inclusive)
filtered=data[(data>=15) & (data<=35)]
print("Numbers between 15 to 35")
print(filtered)
```

Output:

```
Numbers between 15 to 35
2    18
3    25
4    30
dtype: int64
```