



## Academic Year: 2022-23 (EVEN)

Semester & Code: CM-6-I

Class: TYCM-I

Course & Code: Programming with Python (PWP-22616)

Sem: VI

## Question Bank For MSBTE Summer-2023 Practical End Semester Examination

1. Write a Python program to find sum of all elements, largest element and smallest element from the Tuple. Input a tuple from User.  

```
t1=()
cnt=int(input("Enter Count of Tuple:"))
for i in range(cnt):
    ele=int(input(f"Enter {i} Element:"))
    t1+=(ele,)
print(t1)
print("Sum of Tuple Elements=",sum(t1))
print("Largest Element of Tuple=",max(t1))
print("Smallest Element of Tuple=",min(t1))
```
2. Write a Python program to find the repeated items of a tuple. Input a tuple from User.  

```
tpl = ()
cnt = int(input("Enter length of tuple: "))
for i in range(cnt):
    ele = input(f"Enter {i+1} item: ")
    tpl += (ele,)
print("Original:", tpl)

tpl2 = ()
for item in tpl:
    if tpl.count(item) > 1 and item not in tpl2:
        tpl2 += (item,)
print("Repeated items:", tpl2)
```
3. Print the number in words for Example: 1234 => One Two Three Four. Input a tuple from User.  

```
#t=('zero','one','two','three','four','five','six','seven','eight','nine')
t = tuple(input("Enter the elements of the tuple separated by spaces: ").split())
l=[]
num=int(input("Enter a number="))
while num>0:
    r=t[num%10]
    num=int(num/10)
    l.append(r)
l.reverse()
print(l)
```
4. Write a Python program to display Even numbers from a tuple. Input a tuple from User.  

```
tpl = ()
cnt = int(input("Enter length of tuple: "))
for i in range(cnt):
    ele = int(input(f"Enter {i+1} item: "))
    tpl += (ele,)
print("Original:", tpl)
print("Even Numbers:")
for i in tpl:
    if (i%2==0):
        print(i,end=",")
```

5. Write a Python program to create a set, add member(s) in a set and remove one item from set.

```
s1=set()
cnt = int(input("Enter length of set: "))
for i in range(cnt):
    ele = int(input(f"Enter {i+1} item: "))
    s1.add(ele)
print(type(s1))
print("Set Elements:", s1)
rele = int(input(f"Enter item to remove: "))
s1.remove(rele)
print("Set Elements:", s1)
```

6. Write a Python program to perform following operations on set: intersection of sets, union of sets, set difference, symmetric difference, clear a set.

```
s1={1,2,3,4,5}
s2={4,5,6,7,8}
print("Set=",s1)
print("Intersection=",s1&s2) #s1.intersection(s2)
print("Union=",s1|s2)        #s1.union(s2)
c1=s1.difference(s2)    #c1=s1-s2
c2=s2.difference(s1)    #c2=s2-s1
print("Difference=",c2)
print("Symmetric Difference=",s1^s2) #s1.symmetric_difference(s2)
s1.clear()
print(s1)
```

7. Write a Python program to find maximum and the minimum value in a set and to find the length of a set.

```
s1=set()
cnt = int(input("Enter length of set: "))
for i in range(cnt):
    ele = int(input(f"Enter {i+1} item: "))
    s1.add(ele)
print("Maximum Value in a Set=",max(s1))
print("Minimum Value in a Set=",min(s1))
print("Length of a Set=",len(s1))
```

8. Write a Python program to Input elements in the Set and find out only Even Numbers from Set and Display.

```
s1=set()
cnt = int(input("Enter length of set: "))
for i in range(cnt):
    ele = int(input(f"Enter {i+1} item: "))
    s1.add(ele)
print("Even Numbers:")
for i1 in s1:
    if(i1%2==0):
        print(i1,end=" ")
```

9. Input two Sets from User and find the common elements between them without using any inbuilt function.

```
s1=set()
s2=set()
cnt = int(input("Enter length of set-1: "))
for i in range(cnt):
    ele = int(input(f"Enter {i+1} item: "))
    s1.add(ele)
```

```

cnt1 = int(input("Enter length of set-2: "))
for i1 in range(cnt1):
    ele = int(input(f"Enter {i1+1} item: "))
    s2.add(ele)

```

```

print(s1)
print(s2)
print("Common Elements=")
for i2 in s1:
    if i2 in s2:
        print(i2,end=" ")

```

10. Write a Python script to sort (ascending and descending) a dictionary by value.

```

d1={1:14,2:63,3:9}
srt_asc=sorted(d1.values())
srt_desc=sorted(d1.values(),reverse=True)
print(srt_asc)
print(srt_desc)

```

11. Write a Python program to print all even numbers between 1 to 100 using while loop.

```

i=1
while(i<=100):
    if(i%2==0):
        print(i,end=" ")
    i=i+1

```

12. Write a Python program to check if the input year is a leap year or not.

```

def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        print(year, "is a leap year.")
    else:
        print(year, "is not a leap year.")

```

```

year = int(input("Enter a year: "))
is_leap_year(year)

```

13. Write a Python program to print the table of given no.

```

no=int(input("Enter number to print Table:"))
prod=1
for i in range(1,11):
    prod=no*i
    print(prod)

```

14. Write a Python program to print table from 1 to 10 numbers.

```

prod=1
for no in range(1,11):
    for i in range(1,11):
        prod=no*i
        print(prod,end="\t")
    print()

```

15. Write a Python program to check if the input number is prime or not.

```

def prime(no):
    isprime=True
    for i in range(2,no//2):
        if no%i==0:
            isprime=False
            break

```

```

if isprime:
    print(no,"is Prime Number")
else:
    print(no,"isn't Prime Number")

```

```

no=int(input("Enter a Number:"))
prime(no)

```

16. Write a Python program to find the sum of first 10 natural numbers using for loop.

```

sum1=0
for i in range(1,11):
    sum1=sum1+i
print("Sum of first 10 natural numbers =",sum1)

```

17. Write a Python program to print Fibonacci series.

```

def fibonacci(no):
    a=0
    b=1
    for i in range(1,no+1):
        print(a,end=" ")
        next=a+b
        a=b
        b=next
no=int(input("Enter a Number:"))
fibonacci(no)

```

18. Write a Python program to calculate factorial of a number

```

def factorial(no):
    fact=1
    for i in range(1,no+1):
        fact=fact*i
    print("Factorial of",no,"=",fact)
no=int(input("Enter a Number:"))
factorial(no)

```

19. Write a Python Program to Reverse a Given Number/String.

#### **Reversed Number**

```

def reversednum(no):
    rev=0
    ori=no
    while no>0:
        digit=no%10
        rev=(rev*10)+digit
        no=no//10
    print("Reversed of",ori,"=",rev)
no=int(input("Enter a Number:"))
reversednum(no)

```

#### **Reversed String**

```

def reversedstring(string):
    rstring=string[::-1]
    print("Reversed String=",rstring)
string=input("Enter a String:")
reversedstring(string)

```

20. Write a Python program to input a number and find the sum of digits in a number.

```

def sumofdigit(no):
    sum1=0
    while(no>0):

```

```

    digit=no%10
    sum1=sum1+digit
    no=no//10
    print("Sum of digits =",sum1)
no=int(input("Enter a Number:"))
sumofdigit(no)

```

21. Write a Python program that takes a number and checks whether it is a palindrome or not.

```

def is_palindrome(no):
    if str(no)==str(no)[::-1]:
        print(no,"is Palindrome")
    else:
        print(no,"is not Palindrome")
no=int(input("Enter a number:"))
is_palindrome(no)

```

22. Write a Python program to sum all the items in a list. Input a list from User.

```

l1=[]
cnt = int(input("Enter length of List: "))
for i in range(cnt):
    ele = int(input(f'Enter {i+1} item: '))
    l1.append(ele)
print("Sum of List Elements=",sum(l1))

```

23. Write a Python program to multiplies all the items in a list. Input a list from User.

```

l1=[]
mul=1
cnt = int(input("Enter length of List: "))
for i in range(cnt):
    ele = int(input(f'Enter {i+1} item: '))
    l1.append(ele)
for j in l1:
    mul=mul*j
print("Multiplication of List elements:",mul)

```

24. Write a Python program to get the largest number and smallest number from a list. Input a list from User.

```

l1=[]
cnt = int(input("Enter length of List: "))
for i in range(cnt):
    ele = int(input(f'Enter {i+1} item: '))
    l1.append(ele)
print("Maximum Number from List:",max(l1))
print("Minimum Number from List:",min(l1))

```

25. Write a Python program to reverse a list. Input a list from User.

```

l1=[]
cnt = int(input("Enter length of List: "))
for i in range(cnt):
    ele = int(input(f'Enter {i+1} item: '))
    l1.append(ele)
print("List=",l1)
print("Reversed List:",list(reversed(l1)))

```

26. Write a Python program to find common items from two lists. Input a list from User.

```

l1=[]
cnt = int(input("Enter length of List-1: "))
for i in range(cnt):
    ele = int(input(f'Enter {i+1} item: '))

```

```

    l1.append(ele)
print("List-1=",l1)
l2=[]
cnt = int(input("Enter length of List-2: "))
for j in range(cnt):
    ele = int(input(f"Enter {i+1} item: "))
    l2.append(ele)
print("List-2=",l2)
print("Common Elements")
for i1 in l1:
    if i1 in l2:
        print(i1,end=",")

```

27. Write a Python program to select the even items of a list. Input a list from User.

```

l1=[]
cnt = int(input("Enter length of List: "))
for i in range(cnt):
    ele = int(input(f"Enter {i+1} item: "))
    l1.append(ele)
print("List=",l1)
print("Even Number")
for i in l1:
    if i%2==0:
        print(i,end=",")

```

28. Write a Python script to concatenate the following dictionaries to create a new one.

- Sample Dictionary:
  - dic1 = {1:10, 2:20}
  - dic2 = {3:30, 4:40}
  - dic3 = {5:50,6:60}

```

dic1 = {1:10, 2:20}
dic2 = {3:30, 4:40}
dic3 = {5:50,6:60}
concat_dict = {}

```

```

concat_dict.update(dic1)
concat_dict.update(dic2)
concat_dict.update(dic3)

```

```

print("Concatenated Dictionary:", concat_dict)

```

29. Write a Python program to combine two dictionary adding values for common Keys.

- d1 = {'a': 100, 'b': 200, 'c':300}
- d2 = {'a': 300, 'b': 200, 'd':400}
- Sample Data: [{"V": "S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII": "S005"}, {"V": "S009"}, {"VIII": "S007"}]

```

d1 = {'a': 100, 'b': 200, 'c': 300}
d2 = {'a': 300, 'b': 200, 'd': 400}

```

```

combined_dict = {}

```

```

for key in d1:
    if key in d2:
        combined_dict[key] = d1[key] + d2[key]
    else:
        combined_dict[key] = d1[key]

```

```

for key in d2:

```

```
if key not in combined_dict:
    combined_dict[key] = d2[key]
```

```
print("Combined Dictionary:", combined_dict)
```

30. Write a Python program to print all unique values in a dictionary.

```
my_dict = {'a': 100, 'b': 200, 'c': 100, 'd': 300, 'e': 200}
unique_values = set()
for value in my_dict.values():
    unique_values.add(value)
print("Unique values in the dictionary:", unique_values)
```

31. Write a Python program to find the highest 3 values in a dictionary.

```
dict1 = {1: 500, 2: 400, 3: 300, 4: 200, 5: 100}
sorted_items = sorted(dict1.items(), key=lambda x: x[1], reverse=True)
highest_values = sorted_items[:3]
print("Highest 3 values in the dictionary:")
for key, value in highest_values:
    print(f"Key: {key}, Value: {value}")
```

32. Python Program to input values from the user in a dictionary where keys will be automatically generated using range.

```
num_keys = int(input("Enter Length of the number of keys: "))
dict1 = {}
for key in range(1, num_keys+1):
    value=input(f"Enter value for key {key}: ")
    dict1[key] = value
print("Dictionary:",dict1)
```

33. Write a Python program to generate a random float where the value is between 5 and 50 using the Python math module.

```
import random as r
print(r.uniform(5,50))
```

34. Python program to convert given string into another case. If the string is in uppercase, convert to lowercase and vice-versa.

```
def letters(string):
    if (string.islower()):
        cstring=string.upper()
        print(cstring)
    else:
        cstring=string.lower()
        print(cstring)
string=input("Enter a String:")
letters(string)
```

35. Write a Python function that takes a number as a parameter and check whether the number is prime or not.

```
def prime(no):
    isprime=True
    for i in range(2,no//2):
        if no%i==0:
            isprime=False
            break
    if isprime:
        print(no,"is Prime Number")
    else:
        print(no,"isn't Prime Number")
```

```
no=int(input("Enter a Number:"))
prime(no)
```

36. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument.

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
n=int(input("Enter a number to calculate factorial: "))
print("Factorial is :",factorial(n))
```

37. Write a Python function that accepts a string and calculate the number of uppercase letters and lowercase letters.

```
def letters(string):
    ul=0
    ll=0
    for i in string:
        if i.islower():
            ll+=1
        else:
            ul+=1
    print("Upper Case Letter Count=",ul)
    print("Lower Case Letter Count=",ll)
string=input("Enter a string:")
letters(string)
```

38. Write a Python program with the user defined function which accepts a number & returns Fibonacci series of given numbers.

```
def fibonacci(no):
    a=0
    b=1
    for i in range(1,no+1):
        print(a,end=" ")
        next=(a+b)
        a=b
        b=next
no=int(input("Enter a number:"))
fibonacci(no)
```

39. Write a Python program to create a user defined module that will ask your college name and will display the name of the college.

**Module.py**

```
def college(name):
    return name
import
import collegename as c
clg=input("Enter College Name:")
print("College=",c.college(clg))
```

40. Write a Python program to create a class 'Degree' having a method 'getDegree' that prints "I got a degree". It has two subclasses namely 'Undergraduate' and 'Postgraduate' each having a method with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively. Call the method by creating an object of each of the three classes.

```
class Degree:
    def getDegree(self):
        print("I got a degree")
```



```

class Undergraduate(Degree):
    def getDegree(self):
        print("I am an Undergraduate")
class Postgraduate(Degree):
    def getDegree(self):
        print("I am an Postgraduate")
d=Degree()
d.getDegree()
ug=Undergraduate()
ug.getDegree()
pg=Postgraduate()
pg.getDegree()

```

41. Create a class Employee with data members: name, department and salary. Create suitable methods for reading and printing employee information.

```

class Employee:
    def getemp(self):
        self.name=input("Enter Name of Employee:")
        self.department=input("Enter Department of Employee:")
        self.salary=float(input("Enter Salary of Employee:"))
    def putemp(self):
        print("Enter Name of Employee:",self.name)
        print("Enter Department of Employee:",self.department)
        print("Enter Salary of Employee:",self.salary)
emp=Employee()
print("Enter Employee Details\n")
emp.getemp()
print("\nEmployee Details\n")
emp.putemp()

```

42. Python program to read and print students information using two classes using simple inheritance.

```

class College:
    def getclg(self):
        self.cname=input("Enter College Name:")
    def putclg(self):
        print("College Name:",self.cname)
class Student(College):
    def getstud(self):
        self.roll=int(input("Enter Roll No.:"))
        self.name=input("Enter Name:")
    def putstud(self):
        print("Roll No.:",self.roll)
        print("Name:",self.name)
s1=Student()
s1.getclg()
s1.getstud()
s1.putclg()
s1.putstud()

```

43. Write a Python program to implement multiple inheritance assuming suitable data.

```

class Test:
    def getTest(self):
        self.t1=int(input("Enter Test-1 Mark:"))
        self.t2=int(input("Enter Test-2 Mark:"))
    def putTest(self):
        print("Test Marks:", "Test-1=",self.t1, "Test-2=",self.t2)
class MicroProject:
    def getMp(self):

```

```

        self.mpmark=int(input("Enter Microproject Marks:"))
    def putMp(self):
        print("Microproject Marks:",self.mpmark)
class Result(Test,MicroProject):
    def disp(self):
        self.putTest()
        self.putMp()
        self.totaltest=(self.t1+self.t2)/2.0
        self.avg=(self.totaltest+self.mpmark)
        print("Average Total Marks(outoff 30):",self.avg)
r=Result()
r.getTest()
r.getMp()
r.disp()

```

44. Write a Python program to Check for ZeroDivisionError Exception.

```

no1=int(input("Enter Number-1:"))
no2=int(input("Enter Number-2:"))
try:
    print("Division=",no1/no2)
except ZeroDivisionError as ze:
    print("Error Occured:",ze)
else:
    print("Division Performed Successfully..")

```

45. Write a Python program to create user defined exception that will check whether the password is correct or not using direct method.

```

class Myexp(Exception):
    pass
def checkPass(pwd):
    try:
        passwd="aadi2234"
        if pwd==passwd:
            print("Valid Password")
        else:
            raise Myexp("Incorrect Password")
    except Myexp as e:
        print("Error Occurred:",e)
pwd=input("Enter Password:")
checkPass(pwd)

```

46. Write a Python program to create user defined exception that will check whether the username is correct or not using indirect method.

```

class Baseexp(Exception):
    pass
class Myexp(Baseexp):
    pass
def checkPass(username):
    try:
        uname="KKWP"
        if username==uname:
            print("Correct Username")
        else:
            raise Myexp("Incorrect Username")
    except Myexp as e:
        print("Error Occurred:",e)
username=input("Enter Username:")
checkPass(username)

```

47. Write a Python program to create a User Defined Exception to input the age from a person & check whether he/she is an adult, if not raise exception.

```
class PersonExp(Exception):
    pass
def checkAge(age):
    try:
        if age<18:
            print("Person is an Adult")
        else:
            raise PersonExp("Person is not an Adult")
    except ValueError as v:
        print("Plz Enter Valid Age:",v)
    except PersonExp as p:
        print("Error Occurred:",p)

age=int(input("Enter Age:"))
checkAge(age)
```

48. Write a Python program to check for value error & name error exception. Input the data from the user.

```
try:
    no=int(input("Enter a no:"))
    print(no)
except NameError:
    print("No such variable no")
except ValueError:
    print("Only integer value is expected")
finally:
    print("Operation Performed Successfully...")
```

49. Implement a program to copy contents of one file to another but converting them into uppercase.

```
with open('input.txt','r') as src, open('output.txt','w') as dest:
    data=src.read().upper()
    dest.write(data)
    print("UpperCase Data Copied Successfully")
```

50. Implement a program to calculate no. of lines, words and characters in any file.

```
f= open('input.txt')
no_lines=0
no_words=0
no_character=0
for data in f:
    data=data.strip("\n")
    word=data.split()
    no_lines+=1
    no_words+=len(word)
    no_character +=len(data)
print("Lines",no_lines,"words",no_words,"characters",no_character)
```

51. Implement a program to create a new file “first.txt”, write some data in it and show its contents line by line.

```
with open('first.txt','w') as file:
    file.write("Hi..!\nThis is Sample\nPython PR-ESE")
    print("Data Written Successfully..")
with open('first.txt','r') as file:
    for line in file:
        print(line.strip())
```

52. Implement a program to create a new file “first.txt”, write some data in it and show its contents line by

```

line. In this file, append some another data in file and show whether data is updated or not.
with open('first.txt','w') as file:
    file.write("Hi..!\nThis is Sample\nPython PR-ESE")
    print("Data Written Successfully..")
with open('first.txt','r') as file:
    for line in file:
        print(line.strip())
with open('first.txt','a') as file:
    file.write("\nHello..!\nThis is Appended Data\nPython PR-ESE")
    print("Data Appended Successfully..")
with open('first.txt','r') as file:
    for line in file:
        print(line.strip())

```

53. Create a directory named “first\_dir” and apply various directory handling methods in Python on it.

```

import os
os.mkdir('first_dir')
print("Current Working Directory:",os.getcwd())
print("List Directory:",os.listdir())
print('Renamed Directory',os.rename('first_dir','Sample'))
print("Removed Directory",os.rmdir('PWP'))
print("Changed Directory",os.chdir('Practice'))

```

54. Input a list from user and perform all operations on it.

```

l1=[]
l2=['abc','xyz','lmn']
cnt=int(input("Enter Count of List:"))
for i in range(cnt):
    ele=int(input(f'Enter {i} index Element:'))
    l1.append(ele)
print("List Elements:",l1)
print("Accessing List Elements:",l1[2])
l1.insert(1,'PHP')
print("Insert List Elements:",l1)
l1.extend(l2)
print("Extend List Elements:",l1)
l1.pop()
print("Remove List Elements:",l1)

```

55. Input a set from user and perform all operations on it.

```

s1={1,2,3,4,5}
s2={4,5,6,7,8}
print("Set=",s1)
print("Intersection=",s1&s2) #s1.intersection(s2)
print("Union=",s1|s2) #s1.union(s2)
c1=s1.difference(s2) #c1=s1-s2
c2=s2.difference(s1) #c2=s2-s1
print("Difference=",c2)
print("Symmetric Difference=",s1^s2) #s1.symmetric_difference(s2)
s1.clear()
print(s1)

```

56. Input a dictionary from user and perform all operations on it.

```

dict1 = {}
cnt = int(input("Enter Length of Dictionary: "))
for i in range(cnt):
    key = input("Enter a Key: ")
    value = input("Enter a Value: ")

```

```

dict1[key] = value
print("Dictionary:", dict1)
print("Values:", dict1.values())
print("Value for key '2':", dict1.get('2'))
print("Keys:", dict1.keys())
print("Items:", dict1.items())
dict2 = dict1.copy()
print("Popped item from dict2:", dict2.popitem())
print("Popped value for key '3' from dict2:", dict2.pop('3'))
print("dict2 after popping:", dict2)
dict2.clear()
print("dict2 after clearing:", dict2)

```

57. Write a Python program that will calculate area and circumference of circle using inbuilt Math Module.

```

import math as m
radius=float(input("Enter radius of circle:"))
area=m.pi*radius**2
circum=2*m.pi*radius
print("Area of circle:",area)
print("Circumference of circle:",circum)

```

58. Write a Python program that will display Calendar of given month using Calendar Module.

```

import calendar as c
y=int(input("Enter year:"))
m=int(input("Enter Month:"))
print(c.month(y,m))

```

59. Python program to generate random int & float within the given range using random module.

```

import numpy as np
var=np.random.randint(10,30)
print("Integers between 10 to 30:",var)
var2=np.random.uniform(10.5,60.5)
print("Float between 10.5 to 60.5:",var2)

```

60. Use operator module & perform all arithmetic operations on given input.

```

import operator as o
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
print("Addition of number:")
print(o.add(a,b))
print("Subtraction of number:")
print(o.sub(a,b))
print("Multiplication of number:")
print(o.mul(a,b))
print("Division of number:")
print(o.floordiv(a,b))

```

61. Use datetime built-in module & show fulldate, time, day, month and year.

```

from datetime import date
from datetime import datetime
today=date.today()
print("Today's date is:",today)
now=datetime.now()
current_time=now.strftime("%H:%M:%S")
print("Time:",current_time)
print("Year:",today.year)
print("Month:",today.month)
print("Day:",today.day)

```

62. Write a Python program to create two matrices and perform addition, subtraction, multiplication and division operation on matrix.

```
import numpy as np
```

```
matrix1 = np.array([[1, 2, 3],  
                    [2, 1, 2],  
                    [3, 2, 1]])
```

```
matrix2 = np.array([[1, 5, 3],  
                    [2, 6, 7],  
                    [4, 8, 1]])
```

```
add_matrix = np.add(matrix1, matrix2)  
print("Addition=\n", add_matrix)
```

```
sub_matrix = np.subtract(matrix1, matrix2)  
print("Subtraction=\n", sub_matrix)
```

```
mul_matrix = np.multiply(matrix1, matrix2)  
print("Multiplication=\n", mul_matrix)
```

```
div_matrix = np.divide(matrix1, matrix2)  
print("Division=\n", div_matrix)
```

63. Write a Python program to concatenate two strings.

```
str1="Hello "  
str2="Aditya"  
str3=str1+str2  
print("Concatenated string is:",str3)
```

64. Write a NumPy program to generate six random integers between 10 and 30.

```
import numpy as np  
var=np.random.randint(10,30,6)  
print("Random Integers between 10 to 30:",var)
```

65. Write a Python program using numpy Poly1d to generate the polynomial equation and to use det(), inv() on matrices.

```
from numpy import poly1d  
import numpy  
p=poly1d([4,8,9])  
print(p)  
arr=numpy.array([[3,4,2],[5,7,12],[21,33,11]])  
print("Array:\n",arr)  
print("Determinant of matrix:",numpy.linalg.det(arr))  
print("Inverse of matrix:\n",numpy.linalg.inv(arr))
```

66. Design a class Employee with data members: name, department and salary. Create suitable methods for reading and printing employee information.

```
class Employee:  
    def getemp(self):  
        self.name=input("Enter Name of Employee:")  
        self.department=input("Enter Department of Employee:")  
        self.salary=float(input("Enter Salary of Employee:"))  
    def putemp(self):  
        print("Enter Name of Employee:",self.name)  
        print("Enter Department of Employee:",self.department)  
        print("Enter Salary of Employee:",self.salary)
```

```

emp=Employee()
print("Enter Employee Details\n")
emp.getemp()
print("\nEmployee Details\n")
emp.putemp()

```

67. Design a class to show the use of parameterized constructor.

```

class Car:
    def __init__(self,brand,color):
        self.brand=brand
        self.color=color
    def show(self):
        print("Car Brand:",self.brand)
        print("Car Brand:",self.color)
c=Car("Toyota","Black")
c.show()

```

68. Write a Python program to create a class to print an integer and a character with two methods having the same name but different sequence of the integer and the character parameters. For example, if the parameters of the first method are of the form (int n, char c), then that of the second method will be of the form (char c, int n).

```

class Overloading:
    def show(self, n=None, c=None):
        if n and c:
            print("Number=", n, " Character=", c)
        if n is None or c is None:
            print("Insufficient Data")

```

```

o = Overloading()
o.show(n=2, c='A')
o.show(c='Z', n=92)

```

69. Write a Python program to create a class to print the area of a square and a rectangle. The class has two methods with the same name but different number of parameters. The method for printing area of rectangle has two parameters which are length and breadth respectively while the other method for printing area of square has one parameter which is the side of square.

```

class AreaCalculator:
    def calculate_area(self, side, breadth=None):
        if breadth is None:
            area = side ** 2
            print("Area of square:", area)
        else:
            area = side * breadth
            print("Area of rectangle:", area)

```

```

calc = AreaCalculator()

```

```

calc.calculate_area(5)
calc.calculate_area(4, 6)

```