

1.a Develop a program to read the student's details like Name, USN, and Marks in three subjects. Display the student's details, total marks, and percentage with suitable messages.

```
name = input("Enter the Name of student:")
usn = input("Enter the USN of the student:")
m1 = float(input("Enter the marks in first subject:"))
m2= float(input("Enter the marks in second subject:"))
m3 = float(input("Enter the marks in third subject:"))
total_marks = m1+m2+m3
per =(total_marks/300)*100
print("Students details are:")
print("Name is:", name)
print("USN is: ", usn)
print ("Marks in first subject", m1)
print("Marks in second subject", m2)
print ("Marks in third subject", m3)
print ("Total Marks obtained:", total_marks)
print ("Percentage of Marks", per)
```

OUTPUT:

```
Enter the Name of student:Raju
Enter the USN of the student:sg20cse001
Enter the marks in first subject:45
```

Enter the marks in second subject:67

Enter the marks in third subject:89

Students details are:

Name is: Raju

USN is: sg20cse001

Marks in first subject 45.0

Marks in second subject 67.0

Marks in third subject 89.0

Total Marks obtained: 201.0

Percentage of Marks 67.0

B.Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not.

```
name=input("Enter the name of person")
year_of_birth=int(input("enter the birth year"))
age=2025-year_of_birth
print("The age is:",age)
if age>60:
    print("The person is senior citizen")
else:
    print("The person is not senior citizon")
```

OUTPUT:

Enter the name of person

Rahul

enter the birth year

67

The age is: 1955

The person is senior citizen

c. Read N numbers from the console and create a list. Develop a program to print mean, variance, and standard deviation with suitable messages.

```
import math
mylist=[]
n=int(input("Enter the no. of elements"))
print("Enter number in array")
for i in range(int(n)):
    num=input("number")
    mylist.append(int(num))
print("given 'n' number")
print(mylist)
sum=0
for i in mylist:
    sum=sum+i
print(sum)
meanval=0
if(len(mylist)==0):
    print("list is empty")
else:
    meanval=sum/len(mylist)
print(meanval)
mymean=meanval
mylen=len(mylist)
temp=0
for i in range(mylen):
    temp+=(mylist[i]-mymean)*(mylist[i]-mymean)
var=temp/mylen
print("variance",var)
std=math.sqrt(var)
print("standard deviation",std)
```

OUTPUT:

SHARNBASVA UNIVERSIT
Python Programming Lab

Enter the no. of elements5
Enter number in array
number4
number5
number3
number2
number6
given 'n' number
[4, 5, 3, 2, 6]
20
4.0
variance 2.0
standard deviation 1.4142135623730951

2.a. Write a program to demonstrate different numbered data types in Python and perform a different arithmetic operation on numbers in Python.

Source Code:-

```
a = 5
print(a, "is of type", type(a))
a = 2.0
print(a, "is of type", type(a))

num1 = int(input('Enter First number: '))
num2 = int(input('Enter Second number '))
add = num1 + num2
dif = num1 - num2
mul = num1 * num2
div = num1 / num2
floor_div = num1 // num2
power = num1 ** num2
modulus = num1 % num2
print('Sum of ',num1 ,'and' ,num2 ,'is :',add)
print('Difference of ',num1 ,'and' ,num2 ,'is :',dif)
print('Product of ',num1 ,'and' ,num2 ,'is :',mul)
print('Division of ',num1 ,'and' ,num2 ,'is :',div)
print('Floor Division of ',num1 ,'and' ,num2 ,'is :',floor_div)
print('Exponent of ',num1 ,'and' ,num2 ,'is :',power)
print('Modulus of ',num1 ,'and' ,num2 ,'is :',modulus)
```

Output:-

```
5 is of type <class 'int'>
8.5 is of type <class 'float'>
Enter First number: 9
Enter Second number 3
Sum of  9 and 3 is : 12
Difference of  9 and 3 is : 6
Product of 9 and 3 is : 27
Division of  9 and 3 is : 3.0
Exponent of  9 and 3 is : 729
Modulus of  9 and 3 is : 0
```

b. Write a program to create, concatenate and print a string and accessing sub-string from a given string.

P	R	O	G	R	A	M	I	Z	
0	1	2	3	4	5	6	7	8	9
-9	-8	-7	-6	-5	-4	-3	-2	-1	

```
# defining strings in Python

# defining strings in Python
# all of the following are equivalent
str1 = 'Hello'
print('String1=',str1)
str = "Hello"
print(str)
str = ''' Hellow '''
print(str)
# triple quotes string can extend multiple lines
str = """Hello, welcome to
    the world of Python"""
print(str)
str2= 'World!'
print('String2=',str2)
# using +
print('str1 + str2 = ', str1 + str2)
# using *
print('str1 * 3 =', str1 * 3)

#Accessing string characters in Python
str3 = 'programiz'
print('str3 = ', str3)

#first character
print('str3[0] = ', str3[0])
```

```
#last character
print('str3[-1] = ', str3[-1])

#slicing 2nd to 5th character
print('str3[1:5] = ', str3[1:5])

#slicing 6th to 2nd last character
print('str3[5:-2] = ', str3[5:-2])
```

OUTPUT:

```
String1= Hello
Hello
Hellow
Hello, welcome to the world of Python
String2= World!
str1 + str2 = HelloWorld!
str1 * 3 = HelloHelloHello
str3 = programiz
str3[0] = p
str3[-1] = z
str3[1:5] = rogr
str3[5:-2] = am
```

c. Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017

```
import time
import datetime
print("Current date and time: ", datetime.datetime.now())
print("Current year: ", datetime.date.today().strftime("%Y"))
print("Month of year: ", datetime.date.today().strftime("%B"))
print("Week number of the year: ", datetime.date.today().strftime("%W"))
```

```
print("Weekday of the week: ", datetime.date.today().strftime("%w"))
print("Day of year: ", datetime.date.today().strftime("%j"))
print("Day of the month : ", datetime.date.today().strftime("%d"))
print("Day of week: ", datetime.date.today().strftime("%A"))
```

output:

Current date and time: 2021-06-03 15:21:56.139605

Current year: 2021

Month of year: June

Week number of the year: 22

Weekday of the week: 4

Day of year: 154

Day of the month : 03

Day of week: Thursday

**d. Read a multi-digit number (as char) from the console.
Develop a program to print the frequency of each digit with
suitable messages.**

```
s=input("Enter the string\n")
n=len(s)
s=s.lower()
for i in range(n):
    c=1
    if(s[i]!='\0'):
        for j in range(i+1,n):
            if s[i]==s[j]:
                c+=1
            s=s[ : j]+ '\0' +s[j+1 : ]
```



```
print("frequency of ",s[i],"is:",c)
```

OUTPUT:

Enter the string

computer dcience

frequency of c 'is: 3

frequency of o 'is: 1

frequency of m 'is: 1

frequency of p 'is: 1

frequency of u 'is: 1

frequency of t 'is: 1

frequency of e 'is: 3

frequency of r 'is: 1

frequency of 'is: 1

frequency of d 'is: 1

frequency of i 'is: 1

frequency of n 'is: 1

3.a Develop a program to find the largest of three numbers

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):
    largest = num1
elif (num2 >= num1) and (num2 >= num3):
    largest = num2
else:
    largest = num3

print("The largest number is", largest)
```

OUTPUT:

```
Enter first number: 56
Enter second number: 37
Enter third number: 898
The largest number is 898.0
```

b. Develop a program to generate a Fibonacci sequence of length (N). Read N from the console.

```
nterms = int(input("How many terms? "))

n1, n2 = 0, 1
count = 0

if nterms <= 0:

    print("Please enter a positive integer")

elif nterms == 1:
```

```
print("Fibonacci sequence upto",nterms,":")
print(n1)
else:
    print("Fibonacci sequence:")
    while count < nterms:
        print(n1)
        nth = n1 + n2
        n1 = n2
        n2 = nth
        count += 1
```

OUTPUT:

How many terms? 5

Fibonacci sequence:

0

1

1

2

3

- c. Write a program to calculate the factorial of a number.
Develop a program to compute the binomial coefficient
(Given N and R).**

```
def nCr(n, r):

    return (fact(n) / (fact(r)
                      * fact(n - r)))

def fact(n):
    if n == 0:
        return 1
    res = 1

    for i in range(2, n+1):
        res = res * i

    return res

n = int(input("enter the value of n:\n"))
r = int(input("enter the value of r:\n"))
print("Factorial of n number\n",int(fact(n)))
print("binomial coefficient\n",int(nCr(n, r)))
```

OUTPUT:

```
enter the value of n:
5
enter the value of r:
3
Factorial of n number
120
binomial coefficient
10
```

4.a Implementing programs using Functions (Largest number in a list, area of shape)

```
def FindLargest(itr, ele, list1):
```

SHARNBASVA UNIVERSIT
Python Programming Lab

```
    if itr == len(list1):
        print("Largest element in the list is: ", ele)
        return

    if ele < list1[itr]:
        ele = list1[itr]

    # Recursive solution
    FindLargest(itr+1, ele, list1)

    return
```

```
list1 = []
```

```
num = int(input("Enter number of elements in list: "))
```

```
for i in range(1, num + 1):
    ele = int(input("Enter elements: "))
    list1.append(ele)
```

```
FindLargest(0, list1[0], list1)
```

OUTPUT:

```
Enter number of elements in list: 5
Enter elements: 34
Enter elements: 667
Enter elements: 32
Enter elements: 12
Enter elements: 6
Largest element in the list is: 667
```

#area of shape

```
def calculate_area(name):

    name = name.lower()

    # check for the conditions
    if name == "rectangle":
        l = int(input("Enter rectangle's length: "))
        b = int(input("Enter rectangle's breadth: "))

        rect_area = l * b
        print("The area of rectangle is {rect_area}.")

    elif name == "square":
        s = int(input("Enter square's side length: "))

        # calculate area of square
        sqt_area = s * s
        print("The area of square is {sqt_area}.")

    elif name == "triangle":
        h = int(input("Enter triangle's height length: "))
        b = int(input("Enter triangle's breadth length: "))

        # calculate area of triangle
        tri_area = 0.5 * b * h
        print("The area of triangle is {tri_area}.",tri_area)

    elif name == "circle":
        r = int(input("Enter circle's radius length: "))
        pi = 3.14

        # calculate area of circle
        circ_area = pi * r * r
        print("The area of circle is {circ_area}.",circ_area)
```

```
elif name == 'parallelogram':
    b = int(input("Enter parallelogram's base length: "))
    h = int(input("Enter parallelogram's height length: "))
    para_area = b * h
    print("The area of parallelogram is
{para_area}.",para_area)

else:
    print("Sorry! This shape is not available")

# driver code
if __name__ == "__main__" :

    print("Calculate Shape Area")
    shape_name = input("Enter the name of shape whose area
you want to find: ")

    # function calling
    calculate_area(shape_name)
```

OUTPUT:

```
Calculate Shape Area
Enter the name of shape whose area you want to find: triangle
Enter triangle's height length: 4
Enter triangle's breadth length: 3
The area of triangle is {tri_area}. 6.0
```

b. implementing a real-time/technical application using Exception handling (Divide by Zero error, Voter's age validity, student mark range validation)

#Divide by Zero error


```
n=int(input("Enter the value of n:"))
d=int(input("Enter the value of d:"))
c=int(input("Enter the value of c:"))
try:
    q=n/(d-c)
    print("Quotient:",q)
except ZeroDivisionError:
    print("Division by Zero!")
```

OUTPUT:

```
Enter the value of n:8
Enter the value of d:5
Enter the value of c:5
Division by Zero!
```

#Voter's age validity

```
def main():
    #get the age
    try:
        age=int(input("Enter your age"))
        if age>18:
            print("Eligible to vote")
        else:
            print("Not eligible to vote")
    except:
        print("age must be a valid number")
main()
```

OUTPUT:

Enter your age 20
Eligible to vote

Enter your age15
Not eligible to vote

Enter your age eleven
eleven
age must be a valid number

#student mark range validation

```
marks = float(input("Enter your marks in Computer  
Science: "))
```

```
if marks > 90:  
    print("Grade: O")  
elif marks >= 80 and marks < 90:  
    print("Grade: A+")  
elif marks >= 70 and marks < 80:  
    print("Grade: A")  
elif marks >= 60 and marks < 70:  
    print("Grade: B+")  
elif marks >= 50 and marks < 60:  
    print("Grade: B")  
elif marks >= 40 and marks < 50:  
    print("Grade: C")  
else:  
    print("Grade: F")
```

OUTPUT:

Enter your marks in Computer Science: 24
Grade: F

5.a “LIST1” is a list that contain “N” different SRN of students read using a user-defined function with the help of input () function.it is required to add the SRN of “M” for more students that are to be appended or inserted into “LIST1” at the appropriate place. The program must return the index of the SRN entered by the user.

SHARNBASVA UNIVERSIT
Python Programming Lab

```
list = input("Enter elements: \n").split()

# Printign the list.
print("List:\n ", list)

x=input()
list.append(x)

print("After appending the value to list:\n",list)

list.insert(2,"sg18cse011")
print("After inserting the value to the list:\n",list)

index = list.index('sg18cse003')
print("Index value:\n",index)
```

OUTPUT:

Enter elements:

sg18cse001 sg18cse002 sg18cse003 sg18cse004

List:

['sg18cse001', 'sg18cse002', 'sg18cse003', 'sg18cse004']

sg18cse006

After appending the value to list:

['sg18cse001', 'sg18cse002', 'sg18cse003', 'sg18cse004',
'sg18cse006']

After inserting the value to the list:

['sg18cse001', 'sg18cse002', 'sg18cse011', 'sg18cse003',
'sg18cse004', 'sg18cse006']

Index value:

3

5.b TUPLE1” and “TUPLE2” are two tuples that contain “N” values of Different data types read using the user -defined function “READ” with the help of the input () function. Elements of “TUPLE1” and “TUPLE2” are to be read one at a time and the “larger” value among them should be placed into “TUPLE3”. Display all tuples.

```
thistuple = (12,45,22,44,23)
for x in thistuple:
    print(x)
thistuple1 = (1,13,5,67,2)
for y in thistuple1:
    print(y)
abc=thistuple+thistuple1
print("Max of two tuple:\n",abc)
xyz=max(abc)
print(xyz)
```

OUTPUT:

```
12
45
22
```

44

23

1

13

5

67

2

Max of two tuple:

(12, 45, 22, 44, 23, 1, 13, 5, 67, 2)

67

```
thistuple = (12,45,22,44,23)
```

```
for x in thistuple:
```

```
    print(x)
```

```
thistuple1 = (1,13,5,67,2)
```

```
for y in thistuple1:
```

```
    print(y)
```

```
abc=thistuple+thistuple1
```

```
print("Max of two tuple:\n",abc)
```

```
xyz=max(abc)
```

```
print(xyz)
```

```
l=()
```

```
t=list(l)
```

```
t.insert(0,xyz)
```

```
#print(t)
```

```
t_insert=tuple(t)
```

```
print(t_insert)
```

OUTPUT:

12

45

22

44

23

SHARNBASVA UNIVERSIT
Python Programming Lab

1

13

5

67

2

Max of two tuple:

(12, 45, 22, 44, 23, 1, 13, 5, 67, 2)

67

(67,)

6.a SET1 and SET2 are two sets that contain unique integers. SET3 is to be created by taking the union or intersection of SET1 and SET2 using the user defined function Operation (). Perform either union or intersection by reading the choice from the user. Do not use built-in function union () and intersection () and also the operators “|” and “&”.

#UNION

```
set1 = {2, 4, 5, 6,7,8}
set2 = {4, 6, 7, 8,9,10}
set3 = list(set1)
for item in set2:
    if item not in set3:
        set3.append(item)
set3={item for item in set3}

print(set3)
```

OUTPUT:

{2, 4, 5, 6, 7, 8, 9, 10}

#INTERSECTION

```
set1 = {2, 4, 5, 6,7,8}
set2 = { 7, 8,9,10}
set3 = []
for item in set1:
```



```
for num in set2:
    if item==num:
        set3.append(item)
#set3={item for item in set3}

print(set3)
```

OUTPUT:

[7, 8]

6.b. The Dictionary “DICT1” contains N Elements and each element in the dictionary has the operator as the KEY and operands as VALUES. Perform the operations on operands using operators stored as keys. Display the results of all operations.

```
DICT1 = {
    '+': [1, 7, 3],
    '-': [10, 2, 3],
    '*': [2, 3, 4],
    '/': [100, 2, 5]
}
```

```
# Perform operations and display results
for operator, operands in DICT1.items():
    if operator == '+':
        result = sum(operands)
    elif operator == '-':
        result = operands[0] - sum(operands[1:])
    elif operator == '*':
        result = 1
        for operand in operands:
            result *= operand
    elif operator == '/':
        result = operands[0]
        for operand in operands[1:]:
            result /= operand
    print(f"Result of {operator}: {result}")
```

Output:

Result of +: 11

Result of -: 5

Result of *: 24

Result of /: 10.0

7. Implementing programs using Strings. (Reverse, palindrome, character count, replacing characters)

#Reverse # palindrome

```
def reverse(s):
    str = ""
    for i in s:
        str = i + str
    return str
print("Enter the string: ")
s=input()

print("The original string is : ", end="")
print(s)

print("The reversed string(using loops) is : ", end="")
print(reverse(s))
if s==reverse(s):
    print('Is a palindrome')
else:
    print('Is not a palindrome')
```

OUTPUT:

```
Enter the string:
madam
The original string is : madam
The reversed string(using loops) is : madam
Is a palindrome
```

#Character count

```
print("Enter the string: ")
string=input()

count = 0;

for i in range(0, len(string)):
    if(string[i] != ' '):
        count = count + 1;

print("Total number of characters in a string: " + str(count));
```

OUTPUT:

```
Enter the string:
computer science
Total number of characters in a string: 15
```

#replacing characters

```
string = "python is a programming language part"
```

```
# replace "p" with "P" once
s1=string.replace('p', "P",2)
print("After update\n",s1)
```

```
# replace "p" with "P" once
s2=string.replace("p", "P")
print("After update\n",s2)
```

OUTPUT:

```
After update
Python is a Programming language part
After update
Python is a Programming language Part
```


8.a. Develop a program to print the 10 most frequently appearing words in a text file. [Hint: Use dictionary with distinct words and their frequency of occurrences. Sort the dictionary in the reverse order of frequency and display dictionary slice of first 10 items]

```
def print_top_10_words(file_name):
    word_freq = {}
    with open(file_name, 'r') as file:
        text = file.read()
        words = text.split()
    for word in words:
        word = word.lower()
        if word in word_freq:
            word_freq[word] += 1
        else:
            word_freq[word] = 1
    sorted_word_freq = sorted(word_freq.items(), key=lambda x:
x[1], reverse=True)
    for word, freq in sorted_word_freq[:10]:
        print(f"{word}: {freq}")
print_top_10_words("demo.txt")
```

OUTPUT:

Text file: demo.txt

ZEBRA AND OX ARE GOOD FRIENDS. DOGS ARE VERY LOYAL AND
FAITHFUL.

8.b. Develop a program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: String method strip(), list method sort(), append(), and file method open(), readlines(), and write()].

```
def sorting(filename):  
    infile = open(filename)  
    words = []  
    for line in infile:  
        temp = line.split()  
        for i in temp:  
            words.append(i)  
    infile.close()  
    words.sort()  
    outfile = open("result.txt", "w")  
    for i in words:  
        outfile.writelines(i)  
        outfile.writelines(" ")  
    outfile.close()
```

```
sorting("sample.txt")
```

OutPut:

Sample.txt

ZEBRA AND OX ARE GOOD FRIENDS. DOGS ARE VERY LOYAL AND FAITHFUL.

8.c Develop a program to backing up a given Folder (Folder in a current working directory) into a ZIP File by using relevant modules and suitable methods.

```
import os, sys, pathlib, zipfile
dirName = input("Enter Directory name that you want to backup : ")
if not os.path.isdir(dirName):
    print("Directory", dirName, "doesn't exists")
    sys.exit(0)
curDir = pathlib.Path(dirName)
with zipfile.ZipFile("myZip.zip", mode="w") as archive:
    for file_path in curDir.rglob("*"):
        archive.write(file_path,
            arcname=file_path.relative_to(curDir))
if os.path.isfile("myZip.zip"):
    print("Archive", "myZip.zip", "created successfully")
else:
    print("Error in creating zip archive")
```


9. Develop a program that uses class Students which prompts the User to enter marks in three subjects and calculate total marks, Percentage and display the scorecard details. [Hint: Use a list to store the marks in three subjects and total marks. Use_init() method to initialize name, USN and the lists to store marks and total, Use getMarks() method to read marks into the list, and display() method to display the scorecard details.]

```
class students:
    count = 0
    def __init__(self, name):
        self.name = name
        self.marks = []
        students.count = students.count + 1

    def enterMarks(self):
        for i in range(3):
            m = int(input("Enter the marks of %s in %d subject:
"%(self.name, i+1)))
            self.marks.append(m)

    def display(self):
        print (self.name, "got ", self.marks)
    def average(self):
        print(sum(self.marks))
        print(sum(self.marks)/len(self.marks))
name = input("Enter the name of Student:")
s1 = students(name)
s1.enterMarks()
s1.display()
```

```
s1.average()  
print ("")
```

```
name = input("Enter the name of Student:")  
s2 = students(name)  
s2.enterMarks()  
s2.display()  
s2.average()
```

OUTPUT:

```
Enter the name of Student:raju  
Enter the marks of raju in 1 subject: 50  
Enter the marks of raju in 2 subject: 50  
Enter the marks of raju in 3 subject: 50  
raju got [50, 50, 50]  
150  
50.0
```

```
Enter the name of Student:mohan  
Enter the marks of mohan in 1 subject: 60  
Enter the marks of mohan in 2 subject: 60  
Enter the marks of mohan in 3 subject: 60  
mohan got [60, 60, 60]  
180  
60.0
```

10. Implementing program using modules and python Standard Libraries (pandas, Numpy, Matplotlib, Scipy)

10. Implementing program using modules and python Standard Libraries (pandas, Numpy, Matplotlib, Scipy)

```
import pandas as pd
df = pd.DataFrame(
{
    "Name": [ "Braund, Mr. Owen Harris",
    "Allen, Mr. William Henry",
    "Bonnell, Miss. Elizabeth",],
    "Age": [22, 35, 58], "Sex": ["male", "male", "female"],
}
)
print(df)
print(df["Age"])
ages = pd.Series([22, 35, 58], name= "age")
print(ages)
df["Age"].max()
print(ages.max())
print(df.describe())
```

Output

```
Name Age Sex
0 Braund, Mr. Owen Harris 22 male 1
  Allen, Mr. William Henry 35 male 2
  Bonnell, Miss. Elizabeth 58 female 0
22
1 35
2 58
Name: Age, dtype: int64
0 22
1 35
2 58
Name: age, dtype: int64
58
Age
```

```
count 3.000000
mean 38.333333
std 18.230012
min 22.000000
25% 28.500000
```

#10.b. Implementing program using modules and python Standard Library (Numpy)

```
import numpy as np
a = np.arange(6)
a2 = a[np.newaxis, :]
a2.shape
#Array Creation and functions:
a = np.array([1, 2, 3, 4, 5, 6])
a = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
print(a[0])
print(a[1])
np.zeros(2)
np.ones(2)
np.arange(4)
np.arange(2, 9, 2)
np.linspace(0, 10, num=5)
x = np.ones(2, dtype=np.int64)
print(x)
arr = np.array([2, 1, 5, 3, 7, 4, 6, 8])
np.sort(arr)
a = np.array([1, 2, 3, 4])
b = np.array([5, 6, 7, 8])
np.concatenate((a, b))
#Array Dimensions:
array_example = np.array([[[0, 1, 2, 3], [4, 5, 6, 7]], [[0, 1, 2, 3], [4, 5, 6, 7]],
[[0, 1, 2, 3], [4, 5, 6, 7]]])
array_example.ndim
array_example.size
```

```
array_example.shape
a = np.arange(6)
print(a)
b=a.reshape(3, 2)
print(b)
np.reshape(a, newshape=(1, 6), order='C')
```

Output

```
[1 2 3 4]
[5 6 7 8]
[1 1]
[0 1 2 3 4 5]
[[0 1]
 [2 3]
 [4 5]]
```

#10. c. Implementing program using modules and python Standard

Library(Matplotlib)

```
import matplotlib.pyplot as plt
days = [1,2,3,4,5]
sleeping =[7,8,6,11,7]
eating = [2,3,4,3,2]
working =[7,8,7,2,2]
playing = [8,5,7,8,13]
slices = [7,2,2,13]
activities = ['sleeping','eating','working','playing']
cols = ['c','m','r','b'] plt.pie(slices,labels=activities,
colors=cols, startangle=90,
shadow= True,
explode=(0,0.1,0,0),
autopct='%1.1f%%')
plt.title('Pie Plot')
plt.show()
```

Output

Pie Plot

