BE AN ACTioneer, Aspire To BE the BEST

Agni College of Technology

Approved by AICTE / UGC, New Delhi, Accredited by NBA, Affiliated to Anna University, Chennai.
ISO 9001:2015 Certified Institution. Estd. 2001
OMR, Thalambur, Chennai - 600 130. Phone: ✆ 044 6740 9441 Mobile: ✆ 9445024081 www.act.edu.in

# GE3171

# PROBLEM SOLVING AND PYTHON PROGRAMMING

# LABORATORY

# RECORD NOTE BOOK

# (2021-2022)

**Regulation :**

**Branch :**

**Year/ Sem :**

**Name :……..**

**Reg No :……**

# BONAFIDE CERTIFICATE

**Register Number :**

**Name of the lab :** GE3171- Problem Solving and Python Programming Laboratory

**Department :**

Certified that this is a Bonafide Record of Practical Work done by Mr./Ms.……. of ……. Department, First semester in the Problem Solving and Python Programming Laboratory during the year 2021-22.

**Signature of Lab-in-Charge** **Signature of Head of the Department**

Submitted for the University Practical Examination held on ……..

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

**ANNA UNIVERSITY: CHENNAI**

**SYLLABUS (R - 2021)**

## GE3171- PROBLEM SOLVING AND PYTHON PROGRAMMING LABORATORY

**COURSE OBJECTIVES:**

- To understand the problem solving approaches.
- To learn the basic programming constructs in Python.
- To practice various computing strategies for Python-based solutions to real world problems.
- To use Python data structures - lists, tuples, dictionaries.
- To do input/output with files in Python.

**EXPERIMENTS:**

1. Identification and solving of simple real life or scientific or technical problems, and developing flow charts for the same. (Electricity Billing, Retail shop billing, Sin series, weight of a motorbike, Weight of a steel bar, compute Electrical Current in Three Phase AC Circuit, etc.)
2. Python programming using simple statements and expressions (exchange the values of two variables, circulate the values of n variables, distance between two points).
3. Scientific problems using Conditionals and Iterative loops. (Number series, Number Patterns, pyramid pattern)
4. Implementing real-time/technical applications using Lists, Tuples. (Items present in a library/Components of a car/ Materials required for construction of a building -operations of list & tuples)
5. Implementing real-time/technical applications using Sets, Dictionaries. (Language, components of an automobile, Elements of a civil structure, etc.- operations of Sets & Dictionaries)
6. Implementing programs using Functions. (Factorial, largest number in a list, area of shape)
7. Implementing programs using Strings. (reverse, palindrome, character count, replacing characters)
8. Implementing programs using written modules and Python Standard Libraries (pandas, numpy. Matplotlib, scipy)
9. Implementing real-time/technical applications using File handling. (copy from one file to another, word count, longest word)
10. Implementing real-time/technical applications using Exception handling. (divide by zero error, voter's age validity, student mark range validation)
11. Exploring Pygame tool.
12. Developing a game activity using Pygame like bouncing ball, car race etc.

**COURSE OUTCOMES:**

On completion of the course, students will be able to:

**C01:** Develop algorithmic solutions to simple computational problems

**C02:** Develop and execute simple Python programs.

**C03:** Implement programs in Python using conditionals and loops for solving problems. C04: Deploy functions to decompose a Python program.

**C05:** Process compound data using Python data structures.

**C06:** Utilize Python packages in developing software applications.

# INDEX

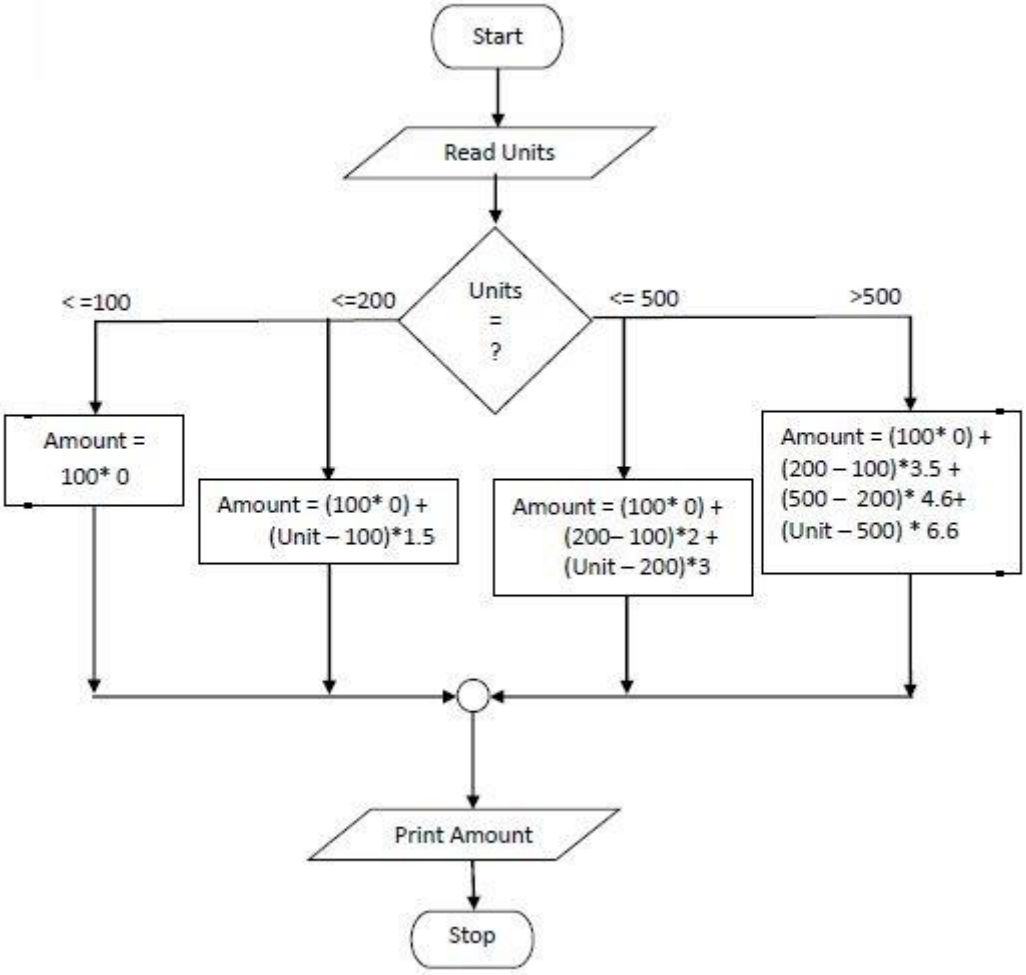
| | | | | |
|---|---|---|---|---|
| | | **CONSTRUCTION OF A BUILDING)** | | |
| **5** | | **OPERATIONS OF SETS & DICTIONARIES (LANGUAGE, COMPONENTS OF AN AUTOMOBILE, ELEMENTS OF A CIVIL STRUCTURE, ETC)** | | |
| **6** | | **PROGRAMS USING FUNCTIONS** | | |
| | | A. Factorial of a Number | | |
| | | B. Largest Number in a list | | |
| | | C. Area of Shape | | |
| **7** | | **PROGRAMS USING STRINGS** | | |
| | | A. Reversing a String | | |
| | | B. Checking Palindrome in a String | | |
| | | C. Counting Characters in a String | | |
| | | D. Replacing Characters in a String | | |
| **8** | | **PROGRAMS USING MODULES AND PYTHON STANDARD LIBRARIES (PANDAS, NUMPY. MATPLOTLIB, SCIPY)** | | |
| **9** | | **PROGRAMS USING FILE HANDLING** | | |
| | | A. Copy from one file to another | | |
| | | B. Word count | | |
| | | C. Longest word | | |
| **10** | | **PROGRAMS USING EXCEPTION HANDLING** | | |

| | | | | |
|---|---|---|---|---|
| | | A. Divide by zero error | | |
| | | B. Voter’s age validity | | |
| | | C. Student mark range validation | | |
| **11** | | **EXPLORING PYGAME TOOL** | | |
| **12** | | **DEVELOPING A GAME ACTIVITY USING PYGAME LIKE BOUNCING BALL, CAR RACE ETC** | | |

| **Ex. No:** 1 A<br><br>**Date:** | **DEVELOPING FLOWCHARTS ELECTRICITY BILLING** |
|---|---|

**Problem Statement:**

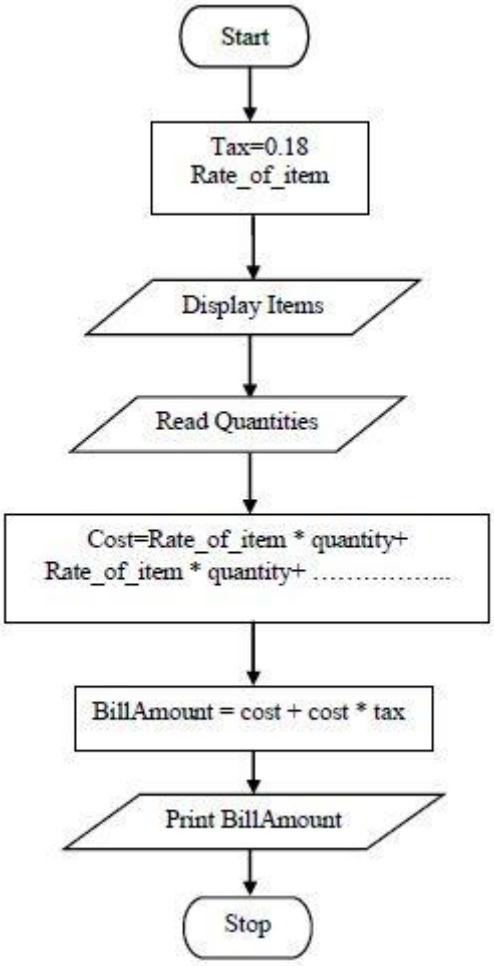
- For 0 to 100 units the per unit is ₹ 0/-
- For 0 to 200 units, for the first 100 unit the per unit cost is zero and the next 100 units, the consumer shall pay ₹ 1.5 per unit.
- For 0 to 500 units, the consumer shall pay ₹ 0 for the first 100 units, for the next 100 units the consumer shall pay ₹ 2 per unit, for the next 300 units the unit cost is ₹3.00/-
- For above 500 units, the consumer shall pay ₹ 0 for the first 100 units, for the next 100 units the consumer shall pay ₹ 3.50 per unit, for the next 300 units the unit cost is ₹4.60/- and for the remaining units the unit cost is ₹6.60/-


Start
Read Units
Units
=
?
< =100
<=200
<= 500
>500
Amount =
100* 0
Amount = (100* 0) +
(Unit – 100)*1.5
Amount = (100* 0) +
(200– 100)*2 +
(Unit – 200)*3
Amount = (100* 0) +
(200 – 100)*3.5 +
(500 – 200)* 4.6+
(Unit – 500) * 6.6
Print Amount
Stop

| **Ex. No:** 1 B<br><br>**Date:** | **RETAIL SHOP BILLING** |
|---|---|

**Problem Statement:**

To prepare Retail shop billing flowchart.

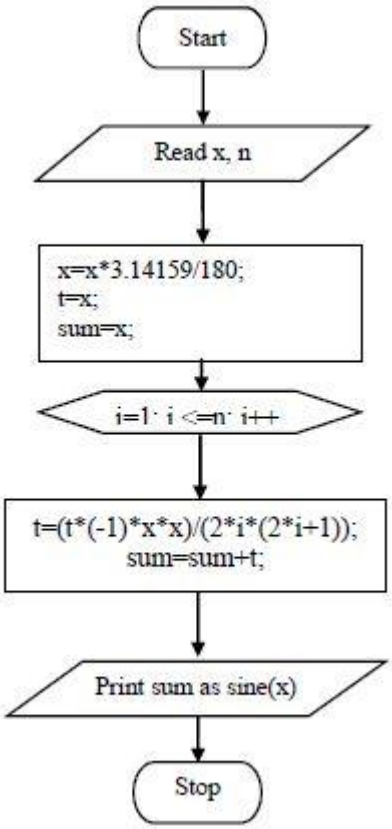


| **Ex. No:** 1 C<br><br>**Date:** | **SINE SERIES** |
|---|---|

**Problem Statement:**

To evaluate the sine series. The formula used to express the Sin(x) as

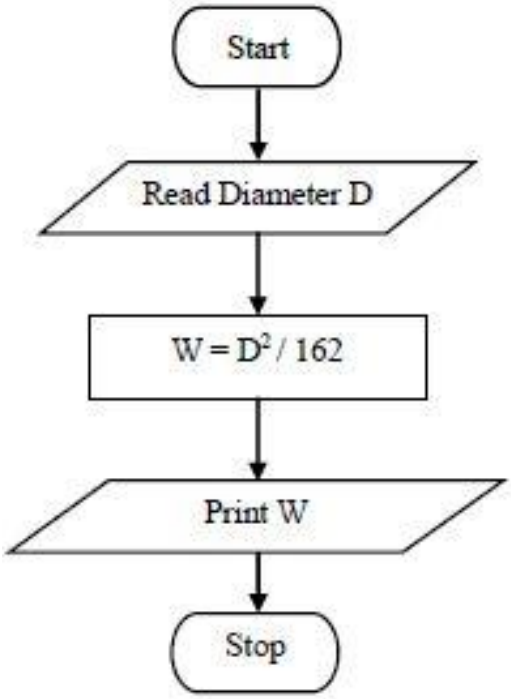
$$\sin x = \sum_{x=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$



| **Ex. No:** 1 D<br><br>**Date:** | **WEIGHT OF A STEEL BAR** |
|---|---|

**Problem Statement:**

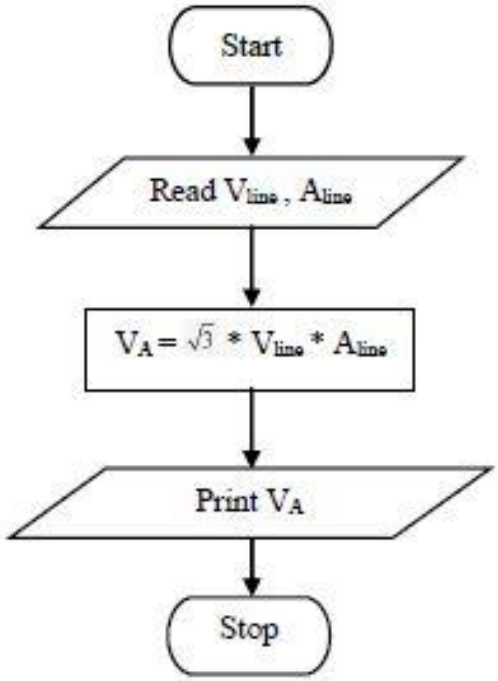
To find the weight of a steel bar.



| **Ex. No:** 1 E<br><br>**Date:** | **COMPUTE ELECTRICAL CURRENT IN THREE PHASE AC CIRCUIT** |
|---|---|

**Problem Statement:**

To compute electrical current in three phase AC circuit.



| | |
|---|---|
| **Ex. No:**2 A<br><br>**Date:** | **PROGRAMS USING SIMPLE STATEMENTS EXCHANGE THE VALUES OF TWO VARIABLES** |

**Aim:**

To write a python program to exchange the values of two variables.

**Algorithm:**

**Program Code:**

```
def exchange(x,y):                                  #Function Definition
        x,y=y,x                                     # swapping using tuple assignment
        print("After exchange of x,y")
        print("x =",x)
        print("Y= ",y)
x=input("Enter value of X ")                        #Main Function
y=input("Enter value of Y ")
print("Before exchange of x,y") print("x =",x)
print("Y= ",y)
exchange(x,y)                                       # Function call
```

**Output:**

```
Enter value of X: 67
Enter value of Y: 56
Before exchange of x,y
x = 67
Y= 56
After exchange of x,y
x = 56
Y= 67
```

| **Ex. No:**2 B<br><br>**Date:** | **CIRCULATE THE VALUES OF N VARIABLES** |
|---|---|

**Aim:**

To write a python program to calculate the values of N variables.

**Algorithm:**

**Program Code:**

```
def circulate(A,N):
    for i in range(1,N+1):
        B=A[i:]+A[:i]
        print("Circulation ",i,"=",B) return
A=[91,92,93,94,95]
N=int(input("Enter n:"))
circulate(A,N)
```

**Output:**

```
Enter n:5
Circulation  1 = [92, 93, 94, 95, 91]
Circulation  2 = [93, 94, 95, 91, 92]
Circulation  3 = [94, 95, 91, 92, 93]
Circulation  4 = [95, 91, 92, 93, 94]
Circulation  5 = [91, 92, 93, 94, 95]
```

| **Ex. No:**2 C<br><br>**Date:** | **DISTANCE BETWEEN TWO VARIABLES** |
|---|---|

**Aim:**

To write a python program to find distance between two variables

**Algorithm:**

**Program Code:**

```
import math
x1 = int(input("Enter a x1: "))
y1 = int(input("Enter a y1: "))
x2 = int(input("Enter a x2: "))
y2 = int(input("Enter a y2: "))
distance = math.sqrt(((x2-x1)**2)+((y2-y1)**2)) print("Distance = ",distance)
```

**Output:**

```
Enter a x1: 3
Enter a y1: 2
Enter a x2: 7
Enter a y2: 8
Distance = 7.211102550927978
```

| **Ex. No:**3 A<br><br>**Date:** | **PROGRAMS USING CONDITIONALS AND ITERATIVE LOOPS NUMBER SERIES** |
|---|---|

**Aim:**

To write a python program to evaluate 12+22+32+….+N2

**Algorithm:**

**Program Code:**

```
n = int(input('Enter a number: '))
sum=0
i=1
while i<=n:
        sum=sum+i*i
        i+=1
print('Sum = ',sum)
```

**Output:**

Enter a number: 10

Sum = 385

| **Ex. No:** 3 B<br><br>**Date:** | **NUMBER PATTERN** |
| --- | --- |

**Aim:**

To write a python program to print number pattern.

**Algorithm:**

**Program Code:**

```
N = 5
for i in range(1,N+1):
        for k in range(N,i, –1):
                print(" ", end =' ')
        for j in range(1,i+1):
                print(j, end =' ')
        for l in range(i-1,0,-1):
                print(l, end =' ')
        print()
```

**Output:**

```
        1
      1 2 1
    1 2 3 2
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
```

| **Ex. No:** 3 C<br><br>**Date:** | **PYRAMID PATTERN** |
| --- | --- |

**Aim:**

To write a python program to print number pattern.

**Algorithm:**

**Program Code:**

```
n = int(input("Enter the number of rows: "))
m = (2 * n) - 2
for i in range(0, n): for j in
    range(0, m):
        print(end=" ")
    m = m - 1  # decrementing m after each loop for j
    in range(0, i + 1):
        # printing full Triangle pyramid using stars
        print("* ", end=' ')
print(" ")
```

**Output:**

```
Enter the number of rows: 9
                *
               * *
              * * *
             * * * *
              * * * * *
             * * * * * *
            * * * * * * *
           * * * * * * * *
          * * * * * * * * *
```

| **Ex. No:** 4A<br><br>**Date:** | **OPERATIONS OF LISTS** |
|---|---|

**Aim:**

To implement operations in a library list.

**Algorithm:**

**Program Code:**

```
 # declaring a list of items in a Library
library = [‘ Books’,’Periodicals’,’Newspaper’,’Manuscripts’,’Maps’,’Prints’,’Documents’,
'Ebooks']
# printing the complete list print('Library:   ',library)
 # printing first element print('first element:  ',library[0])
 # printing fourth element print('fourth element:   ',library[3])
 # printing list elements from 0th index to 4th index print('Items
 in Library from 0 to 4 index:   ',library[0: 5])
 # printing list -7th or 3rd element from the list
 print('3rd or -7th element:  ',library[-7])
# appending an element to the list library.append('Audiobooks')
print('Library list after   append():   ',library)
 # finding index of a specified element
 print('index of \'Newspaper\':   ',library.index('Newspaper'))
 # sorting the elements of iLIst
 library.sort()
 print('after sorting:   ', library);
 # popping an element
 print('Popped elements is:  ',library.pop()) print('after
 pop(): ', library);
 # removing specified element library.remove('Maps')
 print('after removing \'Maps\':   ',library)
 # inserting an element at specified index
 # inserting 100 at 2nd index
 library.insert(2, 'CDs') print('after
 insert:   ', library)
 # Number of Ekements in Library list
```

```
print(' Number of Elements in Library list :  ',library.count('Ebooks'))
```

**Output:**

```
Library:          ['Books', 'Periodicals', 'Newspaper', 'Manuscripts', 'Maps', 'Prints',
'Documents', 'Ebooks']
first element:  Books
fourth element: Manuscripts
 Items in Library from 0 to 4 index:   ['Books', 'Periodicals', 'Newspaper', 'Manuscripts', 'Maps']
3rd or -7th element:  Periodicals
Library list after append():     ['Books', 'Periodicals', 'Newspaper', 'Manuscripts', 'Maps',
'Prints', 'Documents', 'Ebooks', 'Audiobooks']
index of 'Newspaper': 2
after sorting:   ['Audiobooks', 'Books', 'Documents', 'Ebooks', 'Manuscripts', 'Maps',
'Newspaper', 'Periodicals', 'Prints']
Popped elements is:    Prints
after pop(): ['Audiobooks', 'Books', 'Documents', 'Ebooks', 'Manuscripts', 'Maps', 'Newspaper',
'Periodicals']
after removing 'Maps':          ['Audiobooks', 'Books', 'Documents', 'Ebooks', 'Manuscripts',
'Newspaper', 'Periodicals']
after insert:     ['Audiobooks', 'Books', 'CDs', 'Documents', 'Ebooks', 'Manuscripts',
'Newspaper', 'Periodicals']
Number of Elements in Library list :  1
```

| **Ex. No:** 4B<br><br>**Date:** | **OPERATIONS OF TUPLE** |
|---|---|

**Aim:**

To implement operations in a Car Tuple.

**Algorithm:**

**Program Code:**

```
# Python code for various Tuple operation
# declaring a tuple of Components of a car
car = ('Engine','Battery','Alternator','Radiator','Steering','Break','Seat Belt')
# printing the complete tuple print('Components of a car:  ',car)
# printing first element print('first element: ',car[0])
# printing fourth element print('fourth element:  ',car[3])
# printing tuple elements from 0th index to 4th index print('Components of a car from 0 to 4
index:   ',car[0: 5])
# printing tuple -7th or 3rd element from the list print('3rd or -7th element: ',car[-7])
 # finding index of a specified element
print('index of \'Alternator\':  ',car.index('Alternator'))
# Number of Elements in car tuple
print(' Number of Elements in Car Tuple :   ',car.count('Seat Belt'))
#Length of car tuple
print(' Length of Elements in Car Tuple :  ',len(car))
```

**Output:**

```
Components of a car:  ('Engine', 'Battery', 'Alternator', 'Radiator', 'Steering', 'Break', 'Seat
Belt')
first element    :  Engine
fourth element  :          Radiator
Components of a car from 0 to 4  index: ('Engine', 'Battery', 'Alternator', 'Radiator', 'Steering')
3rd or -7th element: Engine index of 'Alternator':      2
Number of Elements in Car Tuple :    1
Length of Elements in Car Tuple :     7
```

| **Ex. No:** 5A<br><br>**Date:** | **OPERATIONS OF SETS** |
|---|---|

**Aim:**

To implement operations in a set using Components of a Language as example.

**Algorithm:**

**Program Code:**

```
L1 = {'Pitch', 'Syllabus', 'Script', 'Grammar', 'Sentences'};
L2 = {'Grammar', 'Syllabus', 'Context', 'Words', 'Phonetics'};
# set union
print("Union of L1 and L2 is  ",L1 | L2)
# set intersection
print("Intersection of L1 and L2 is  ",L1 & L2)
# set difference
print("Difference of L1 and L2 is  ",L1 - L2)
# set symmetric difference
print("Symmetric difference of L1 and L2 is ",L1 ^ L2)
```

**Output:**

```
Union of L1 and L2 is{'Words', 'Pitch', 'Sentences', 'Phonetics', 'Script', 'Grammar', 'Syllabus', 'Context'}
Intersection of L1 and L2 is  {'Grammar', 'Syllabus'} Difference of L1 and L2 is  {'Script', 'Pitch', 'Sentences'}
Symmetric difference of L1 and L2 is  {'Words', 'Context', 'Script', 'Pitch', 'Sentences', 'Phonetics'}
```

| **Ex. No:** 6A<br><br>**Date:** | **FACTORIAL OF A NUMBER USING FUNCTION** |
|---|---|

**Aim:**

To write a python program to find the factorial of a number using functions.

**Algorithm:**

**Program Code:**

```
def fact(n):
        if n = = 1:
         return n else:
        return n*fact(n–1)
num = int(input("Enter a number: "))
print("The factorial of",num,"is",fact(num))
```

**Output:**

```
Enter a number: 5
The factorial of 5 is 120
```

| **Ex. No:** 6B<br><br>**Date:** | **FINDING LARGEST NUMBER IN A LIST USING FUNCTION** |
|---|---|

**Aim:**

To write a python program to find the largest number in a list using functions.

**Algorithm:**

**Program Code:**

```
def myMax(list1):
    print("Largest element is:", max(list1))


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)
print("Largest element is:", myMax(list1))
```

**Output:**

```
Enter number of elements in list: 6
Enter elements: 58
Enter elements: 69
Enter elements: 25
Enter elements: 37
Enter elements: 28
Enter elements: 49 Largest element is: 69
```

| **Ex. No:** 6 C<br><br>**Date:** | **FINDING AREA OF A CIRCLE USING FUNCTION** |
|---|---|

**Aim:**

To write a python program to find the area of a circle using functions.

**Algorithm:**

**Program Code:**

```
def findArea(r):
    PI = 3.142
    return PI * (r*r);


num=float(input("Enter r value:"))
print("Area is %.6f" % findArea(num));
```

**Output:**

```
Enter r  value: 8
Area is 201.088000
```

| **Ex. No:** 7A<br><br>**Date:** | **REVERSING A STRING** |
| --- | --- |

**Aim:**

To write a python program to reverse a string.

**Algorithm:**

**Program:**

```
def reverse(string):
    string = "".join(reversed(string))
    return string


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


print ("The reversed string(using reversed) is : ",end="")
print (reverse(s))
```

**Output:**

```
Enter any string: Python
The original string  is : Python
The reversed string(using reversed) is : nohtyP
```

| **Ex. No:** 7B<br><br>**Date:** | **CHECKING PALINDROME IN A STRING** |
|---|---|

**Aim:**

To write a python program to check palindrome in a string.

**Algorithm:**

**Program:**

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

string = string.casefold()

 rev_string = reversed(string)

if list(string) = = list(rev_string):

      print("It is palindrome")

else:


      print("It is not palindrome")
```

**Output:**

```
Enter string: Python
It is not palindrome

Enter string: madam
It is palindrome
```

| **Ex. No:** 7C<br><br>**Date:** | **COUNTING CHARACTERS IN A STRING** |
|---|---|

**Aim:**

To write a python program to count number of characters in a string.

**Algorithm:**

**Program:**

```
string = input("Enter any string: ")
char = input("Enter a character to count: ")
val = string.count(char)
print(val,"\n")
```

**Output:**

```
Enter any string: python programming Enter
a character to count: n
2
```

| **Ex. No:** 7 D<br><br>**Date:** | **REPLACE CHARACTERS IN A STRING** |
|---|---|

**Aim:**

To write a python program to replace characters in a string.

**Algorithm:**

**Program:**

```
string = input("Enter any string: ")

str1 = input("Enter old string: ")

str2 = input("Enter new string: ")

print(string.replace(str1, str2))
```

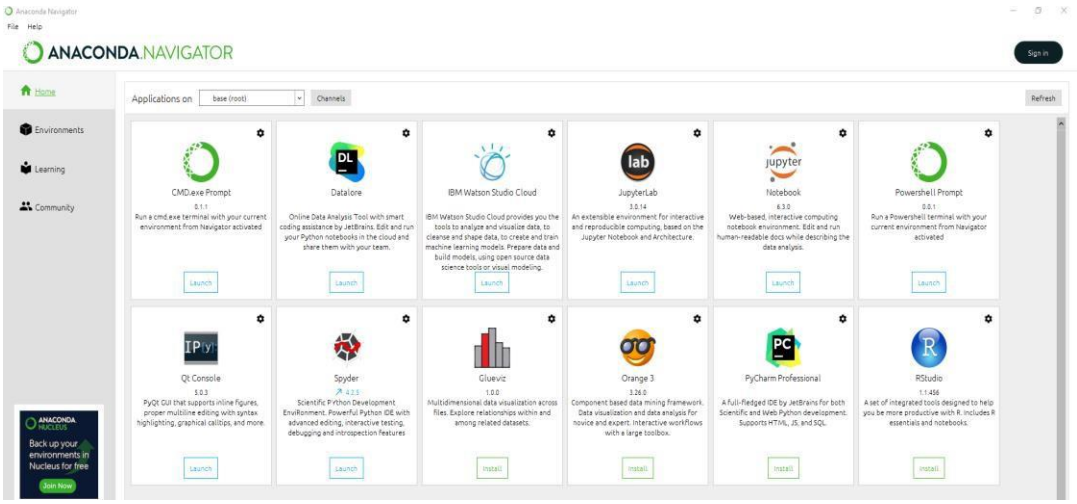
**Output:**

```
Enter any string: problem solving python programming

Enter old string: python

Enter new string: C

problem solving C programming
```

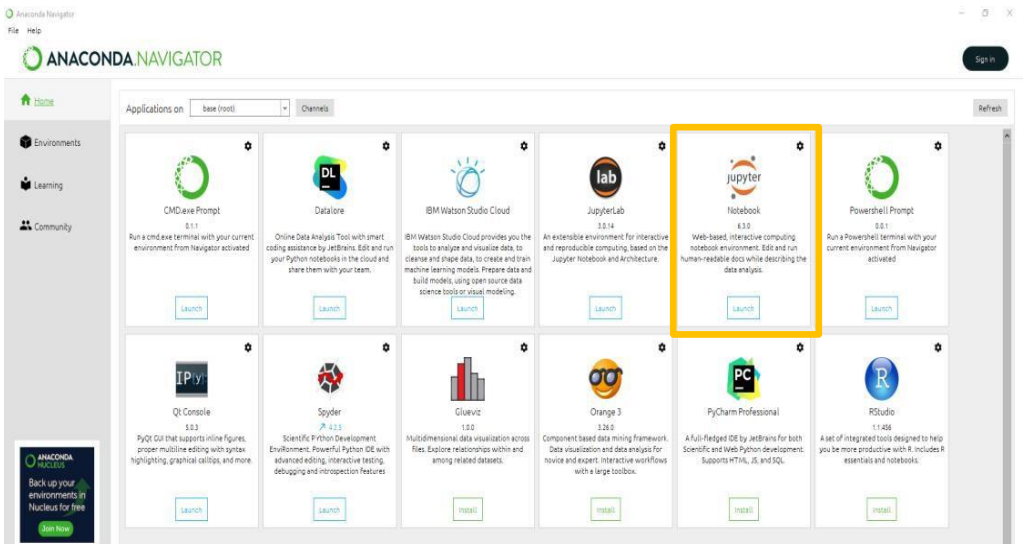
| Ex. No: 8<br><br>Date: | INSTALLING AND EXECUTION PYTHON PROGRAM IN ANACONDA NAVIGATOR<br>(To run pandas, numpy, matplotlib, scipy |
|---|---|

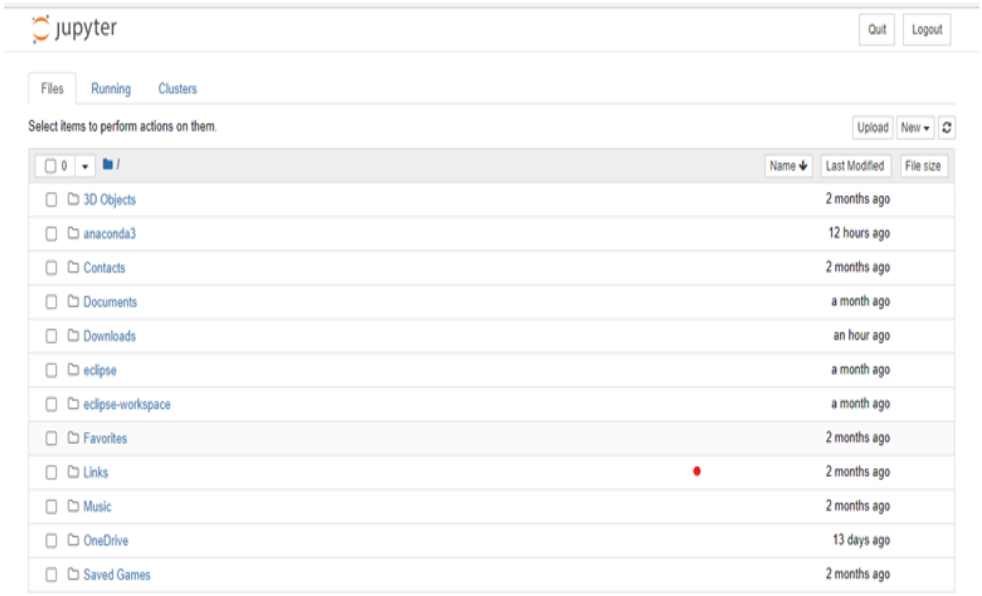
**Step 1:** Install anaconda individual edition for windows

**Step 2:** Open Anaconda navigator



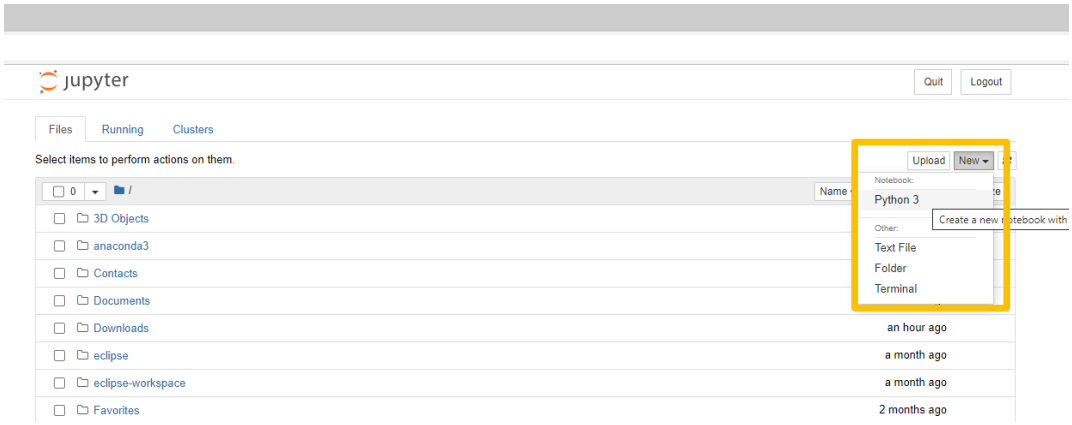

**Step 3:** launch jupyter note book



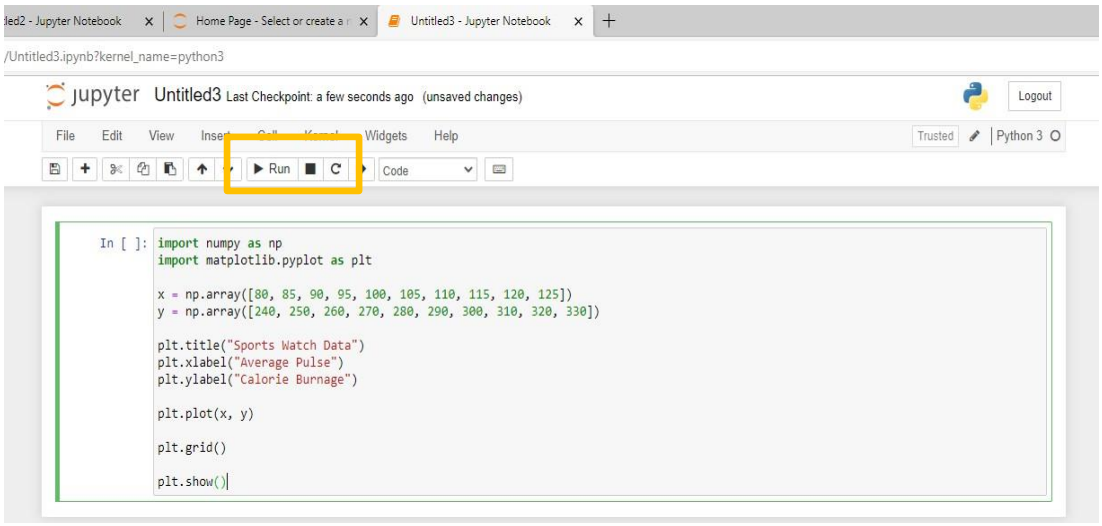



**Step 4:**

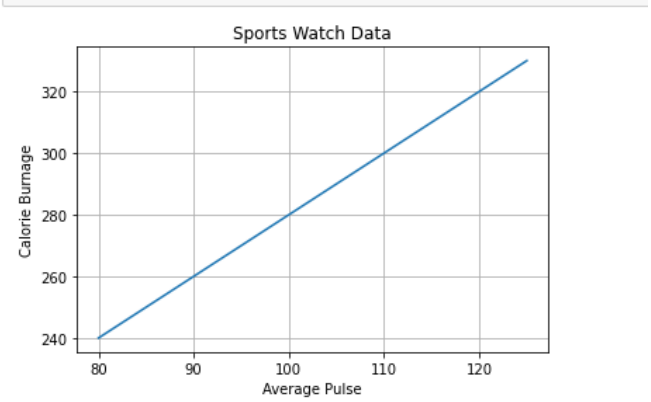
Click new ➞ Python3



**Step 5:** Write or paste the python code




**Step 6:** click run




In [ ]:

| **Ex. No:** 8A<br><br>**Date:** | **PANDAS** |
|---|---|

**Aim:**

To write a python program to compare the elements of the two Pandas Series using Pandas library.

Sample Series: [2, 4, 6, 8, 10], [1, 3, 5, 7, 10]

**Algorithm:**

**Program:**

```
import pandas as pd
ds1 = pd.Series([2, 4, 6, 8, 10])
ds2 = pd.Series([1, 3, 5, 7, 10])
print("Series1:")
print(ds1)
print("Series2:")
print(ds2)
print("Compare the elements of the said Series:")
print("Equals:")
print(ds1 == ds2) print("Greater than:")
print(ds1 > ds2) print("Less than:")
print(ds1 < ds2)
```

**Output:**

```
Series1:
0     2
1     4
2     6
3     8
4    10
dtype: int64
Series2:
0     1
1     3
2     5
3     7
4    10
dtype: int64
Compare the elements of the said Series:
Equals:
0    False
1    False
2    False
3    False
4     True
dtype: bool
Greater than:
0     True
1     True
2     True
3     True
4    False
dtype: bool
Less than:
0    False
1    False
2    False
3    False
4    False
dtype: bool
```

| **Ex. No:** 8B<br><br>**Date:** | **NUMPY** |
|---|---|

**Aim:**

To write a program to test whether none of the elements of a given array is zerousing NumPy library.

**Algorithm:**

**Program:**

```
import numpy as np
x = np.array([1, 2, 3, 4])
print("Original array:")
print(x)
print("Test if none of the elements of the said array is zero:")
print(np.all(x))
x = np.array([0, 1, 2, 3])
print("Original array:")
print(x)
print("Test if none of the elements of the said array is zero:")
print(np.all(x))
```

**Output:**

```
Original array: [1 2 3 4]
Test if none of the elements of the said array is zero: True
Original array: [0 1 2 3]
Test if none of the elements of the said array is zero: False
```

| **Ex. No:** 8C<br><br>**Date:** | **MATPLOTLIB** |
| --- | --- |

**Aim:**

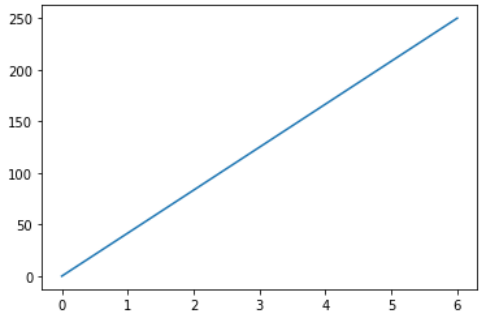
To write a python program to plot a graph using matplotlib library.

**Algorithm:**

**Program:**

```
import matplotlib.pyplot as plt

import numpy as np


xpoints = np.array([0, 6]) ypoints

= np.array([0, 250])


plt.plot(xpoints, ypoints)

plt.show()
```

**Output:**



| **Ex. No:** 8 D<br><br>**Date:** | **SCIPY** |
|---|---|

**Aim:**

To write a python program to return the specified unit in seconds (e.g. hour returns 3600.0) using scipy library.

**Algorithm:**

**Program:**

```
from scipy import constants
print(constants.minute)
print(constants.hour)
print(constants.day)
print(constants.week)
print(constants.year)
print(constants.Julian_year)
```

**Output:**

```
60.0
3600.0
86400.0
604800.0
31536000.0
31557600.
31557600.0
```

| **Ex. No:** 9 A<br><br>**Date:** | **COPY FROM ONE FILE TO ANOTHER** |
|---|---|

**Aim:**

To write a python program to copy from one file to another.

**Algorithm:**

**Program:**

```
from shutil import copyfile
sourcefile = input("Enter source file name: ")
destinationfile = input("Enter destination file name: ")
copyfile(sourcefile, destinationfile)
print("File copied successfully!")
c = open(destinationfile, "r") print(c.read())
c.close()
print()
print()
```

**Output:**

```
Enter source file name: file1.txt
Enter destination file name: file2.txt
File copied successfully!
Sunflower
Jasmine
Roses
```

| **Ex. No:** 9 B<br><br>**Date:** | **WORD COUNT FROM A FILE** |
|---|---|

**Aim:**

To write a python program to count number of words in a file.

**Data.txt**

A file is a collection of data stored on a secondary storage device like hard disk. They can be easily retrieved when required. Python supports two types of files. They are Text files & Binary files.

**Algorithm:**

**Program:**

```
file = open("F:\Data.txt", "rt")
data = file.read()
words = data.split()

print('Number of words in text file :', len(words))
```

**Output:**

```
Number of words in text file : 36
```

| **Ex. No:** 9 C<br><br>**Date:** | **FINDING LONGEST WORD IN A FILE** |
|---|---|

**Aim:**

To write a python program to find longest word in a file

**Data.txt**

A file is a collection of data stored on a secondary storage device like hard disk.They can be easily retrieved when required. Python supports two types of files. They are Text files & Binary files.

**Algorithm:**

**Program:**

```
def longest_word(filename):
    with open(filename, 'r') as infile:
            words = infile.read().split()
    max_len = len(max(words, key=len))
    return [word for word in words if len(word) == max_len]
print(longest_word('F:\Data.txt'))
```

**Output:**

```
['collection']
```

| **Ex. No:** 10 A<br><br>**Date:** | **DIVIDE BY ZERO ERROR USING EXCEPTION HANDLING** |
|---|---|

**Aim:**

To write a python program to handle divide by zero error using exception handling.

**Algorithm:**

**Program:**

```
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:10
Enter the value of d:5
Enter the value of c:5
Division by Zero!
```

| **Ex. No:** 10 B<br><br>**Date:** | **VOTERS AGE VALIDITY** |
| --- | --- |

**Aim:**

To write a python program to check voters age validity.

**Algorithm:**

**Program:**

```
import datetime
Year_of_birth = int(input("In which year you took birth:- "))
current_year = datetime.datetime.now().year
Current_age = current_year - Year_of_birth
print("Your current age is ",Current_age)

if(Current_age<=18):
    print("You are not eligible to vote")
else:
    print("You are eligible to vote")
```

**Output:**

```
In which year you took birth:- 1981 Your
current age is  40
You are eligible to vote

In which year you took birth:- 2011 Your
current age is  10
You are not eligible to vote
```

| **Ex. No:** 10 C<br><br>**Date:** | **STUDENT MARK RANGE VALIDATION** |
|---|---|

**Aim:**

To write a python program to perform student mark range validation

**Algorithm:**

**Program:**

```
Mark = int(input("Enter the Mark: ")) if
Mark < 0 or Mark > 100:
    print("The value is out of range, try again.") else:
    print("The Mark is in the range")
```

**Output:**

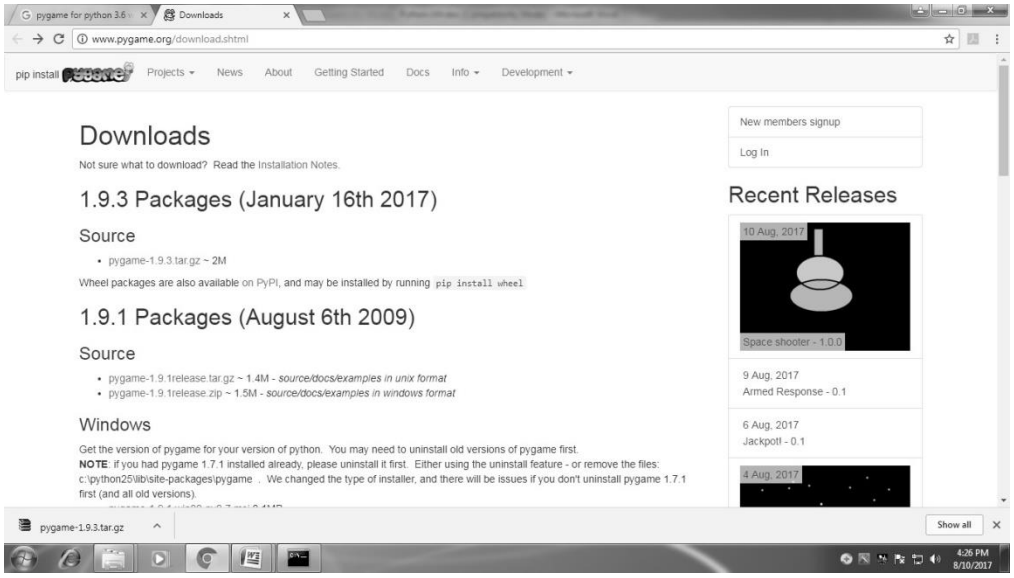
```
Enter the Mark: 150
The value is out of range, try again.

Enter the Mark: 98
The Mark is in the range
```

| Ex. No: 11<br><br>Date: | EXPLORING PYGAME |
|---|---|

# PYGAME INSTALLATION

**To Install Pygame Module Steps**

1. Install python 3.6.2 into C:\
2. Go to this link to install pygame www.pygame.org/download.shtml




3. Click

   pygame-1.9.3.tar.gz ~ 2M and download zar file

4. Extract the zar file into C:\Python36-32\Scripts folder
5. Open command prompt
6. Type the following command

C:\>py -m pip install pygame --user Collecting
pygame
Downloading pygame-1.9.3-cp36-cp36m-win32.whl (4.0MB)

100% |████████████████████████████████| 4.0MB
171kB/s

Installing collected packages: pygame Successfully
installed pygame-1.9.3

7. Now, pygame installed successfully
8. To see if it works, run one of the included examples in pygame-1.9.3

- Open command prompt
- Type the following

C:\>cd Python36-32\Scripts\pygame-1.9.3
C:\Python36-32\Scripts\pygame-1.9.3>cd examples
C:\Python36-32\Scripts\pygame-1.9.3\examples>aliens.py
C:\Python36-32\Scripts\pygame-1.9.3\examples>

| **Ex. No:** 12<br><br>**Date:** | **SIMULATE BOUNCING BALL USING PYGAME** |
|---|---|

**Aim:**

To write a Python program to bouncing ball in Pygame.

**Algorithm:**

**Program:**

```
import pygame pygame.init()
window_w = 800
window_h = 600

white = (255, 255, 255)
black = (0, 0, 0)

FPS = 120

window = pygame.display.set_mode((window_w, window_h))
pygame.display.set_caption("Game: ")
clock = pygame.time.Clock()

def game_loop():
    block_size = 20
    velocity = [1, 1]
    pos_x = window_w/2
    pos_y = window_h/2
    running = True
    while running:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
      pos_x += velocity[0]
      pos_y += velocity[1]

     if pos_x + block_size > window_w or pos_x < 0:
       velocity[0] = -velocity[0]

    if pos_y + block_size > window_h or pos_y < 0:
      velocity[1] = -velocity[1]

# DRAW
window.fill(white)
pygame.draw.rect(window, black, [pos_x, pos_y, block_size, block_size])
pygame.display.update()
clock.tick(FPS)

game_loop()
```