Basic Python Programming

****

Write Python programs to perform the following:

(SIMPLE ARITHMETIC PROGRAMS)

1. Add two numbers.
2. Find out the area and perimeter of a rectangle.
3. Input three decimal numbers and find their sum and average.
4. Input two numbers and swap them.
5. Input temperature in Celsius and convert it to Fahrenheit.

(IF-ELSE & IF-ELSE-IF LADDER)

6. Input a number and find its absolute value.
7. Input a number and check whether it is odd or even and display accordingly.
8. Find the largest and smallest among three numbers supplied by user.
9. Check whether an input year is a leap year or not.
10. Compute the telephone bill for Mr. X as per the call rates given below:
    Rental = 250

1st 100 calls	@Rs. 0.2
Next 100 calls	@ Rs. 0.3
Remaining calls	@ Rs. 0.5

11. Solve a given quadratic equation. [Without imaginary roots].

(LOOP CONTROL STRUCTURE)

12. Print the following patterns up to n no. of lines:

(a)

```
*
* *
* * *
* * * *
* * * * *
```

(b)

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

(c)

```
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

(d)

```
      *
    * * *
  * * * * *
* * * * * * *
  * * * * *
    * * *
      *
```

(e)

```
* * * * * * *
  * * * * *
    * * *
      *
    * * *
  * * * * *
* * * * * * *
```

13. Input two numbers and find their hcf and lcm
14. Input a number n and find:
    (a) Fibonacci series up to n
    (b) the nth Fibonacci number
    (c) First n Fibonacci numbers
15. Input a number and find the sum of its digits using while/do-while loop.
16. Input a number and reverse it using while/do-while loop.
17. Input a number and check if it is a prime number or not.
18. According to the Goldbach conjecture, every even number greater than two is the sum of two prime numbers. Input an even number and decompose it into two primes.
19. Input a number and check whether it is an automorphic number or not using while/do-while loop.
20. Input a number and check whether it is an Armstrong number or not using while/do-while loop.

(FUNCTIONS & RECURSION)

21. Implement simple arithmetic calculator using user defined functions for each operation (addition, subtraction, multiplication, division, modulus, exponent). You may use a dictionary to print the menu.
22. Input a number n and find its factorial using a user defined function long int fact(int)
23. Input a number and check if it a Krishnamurthy number.
24. Find the sum of first n prime numbers using as user defined function to check for prime. Input the value of n from the user.
25. Input a limit n and print all prime fibonacci numbers up to n using a user defined function int prime(int) which returns a 1 if the argument is a prime or else 0.
26. Input a limit n and print all twin prime numbers up to n.
27. Input the values of two variables n and r and calculate ^{n}Cr
28. Generate a pascal’s triangle upto n rows (value of n is to be taken as input from the user).
29. Input a number n and print its prime factors using a user defined function int prime(int) which returns a 1 if the argument is a prime or else 0.
30. Calculate the below series. Use a user defined function fact(n) which will return the factorial of n.
    x/1! – x^3/3! + x^5/5! – x^7/7! + ... (up to n terms)
31. Print the sum of natural numbers up to n using recursion.
32. Find the factorial of a number using recursion.
33. Find the nth Fibonacci number using recursion. The value of n should be taken as input.
34. Find the nth Lucas number using recursion. The value of n should be taken as input.
35. Input two numbers a and b and calculate a^b using recursion.
36. Input two numbers and find their GCD using recursion.
37. Solve the Tower of Hanoi problem for 3 discs using recursion.
38. Solve the Ackermann function for two non-negative integers m and n.

(LISTS & TUPLES)

39. Create a tuple with different data types, print the tuple and then add one more item into the tuple and print it again.
40. Create a tuple and find the repeated items of the tuple.
41. Reverse a given tuple.
42. Find the sum of digits of a number using an user defined function sumofdigits(x) which returns a tuple containing the digits of the number.
43. Input a number and check if it a pefect number or not using an user defined function factors(x) which returns a tuple containing the factors of the number.
44. Convert a given list into a tuple.
45. Create a list of n tuples and sort them according to the second element of each tuple.

46. Create a list containing n tuples and replace last value of all tuples in the list.
    Sample list: [(10,20,40), (40,50,60), (70,80,90)]
    Expected Output: [(10, 20, 100), (40, 50, 100), (70, 80, 100)]
47. Create a list containing n tuples and remove all empty tuple(s) from the list.
    Sample data: [(),(), (",), ('a', 'b'), ('a', 'b', 'c'), ('d')]
    Expected output: [(",), ('a', 'b'), ('a', 'b', 'c'), 'd']
48. Input a list and split the odd and even numbers into two separate lists.
49. Enter the roll, name and marks of n students print their records. Finally find the student name with the highest marks.

(STRINGS)

50. Input two strings and concatenate them.
51. Input a string and reverse it.
52. Enter a sentence and find number of vowels, consonants, spaces and special characters.
53. Input a string and replace each character by the character two place ahead of it, for e.g., a by c, b by d, z by b
54. Input a word and print it vertically.
55. Input a string and check if it a palindrome or not.
56. Input a string and count the number of words in it.
57. Input a string and reverse it using recursion.
58. Input a sentence and find the number of words starting with ‘S’.
59. Input a string and count number of palindromic words present in it using an user defined function palin(word) which returns 1 if the argument is palindrome.
60. Input a name and find its initial (e.g., Subhash Chandra Bose should be printed as S. C. B).
61. Input a name and find its initial (e.g., Subhash Chandra Bose should be printed as S. C. Bose).
62. Input a string and delete all consecutive occurrences of characters.
63. Input a string and a pattern and count the occurrences of the pattern in the string.
64. Input a string and a pattern and delete all occurrences of the pattern from the string.
65. Input a string and two patterns pattern1 and pattern2. Find all occurrences of pattern1 and replace them by pattern2.
66. A Pig Latin word is a word that begins with consonant sound; all letters before the initial vowel are placed at the end of the word sequence. Then, "ay" is added, as in the following examples:
    "pig" "igpay"
    “banana" "ananabay"
     "trash" "ashtray"
    Input a word and generate its Pig Latin.
67. Find the frequency of occurrence of each character in a given string.

(CLASSES & OBJECTS)

68. Create a class called Rectangle having two attributes – length and breadth and find the area and perimeter of the rectangle using two methods – showarea() and showperimeter()
69. Create a class called Complex having two attributes – real and imag. Create two instances of the class and perform addition of two complex numbers. The class should use the following methods:
    show() – that displays the complex numbers in proper format
    add() – that adds the two complex instances
    sub() – that subtracts one complex instance from another
    displaymodulus() – that displays the modulus of the resultant complex number after addition or subtraction
70. Design a class to represent a bank account. Include the following members:
    Data members - Name of the depositor, Account number, Type of account, Account balance
    Methods – To deposit an amount, To withdraw an amount after checking balance, to display the
    name and balance.
    Incorporate a constructor to provide the initial values.

(NUMPY ARRAYS – 1D & 2D)

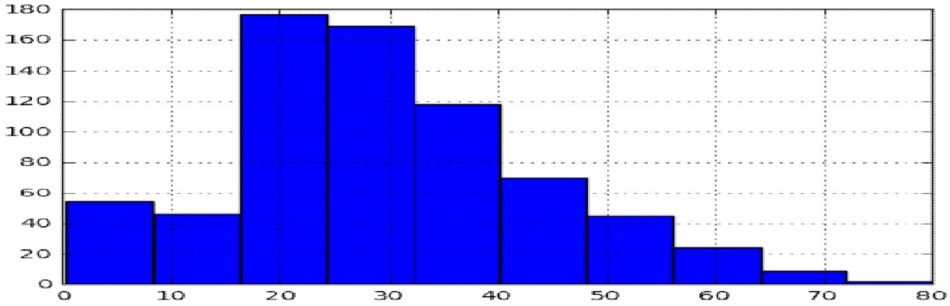
71. Input n random numbers between 0 and 1 and find their sum and average.
72. Input an array of size n and search an item from it. The item should be taken as input from the user and if found, the position at which it has been found in the array should be displayed, else, display "Sorry, item not found."
73. Input an array of n random numbers within 100 and find the maximum and minimum among them.
74. Input an array of n integer elements and find the second highest among them.
75. Input an array of n numbers and store the even numbers into an array even and odd numbers into an array odd. Also display the count of even and odd numbers found.
76. Calculate the mean, median, mode, standard deviation & variance of an array of integers.
77. Input an array of n elements remove all duplicate elements and print the new array.
78. Input two arrays containing m and n elements each (m may or may not be equal to n) in sorted order and merge them into a third array in sorted order and display the merged array.
79. Input two arrays and find the resultant array after performing their –
    (a) Intersection; (b) Union
80. Input an array of n elements in sorted order and perform binary search on it.
81. Input an array of n elements and sort them using –
    (a) bubble sort; (b) selection sort; (c) insertion sort
82. Generate an array of length n whose every odd position is filled with numbers from the Fibonacci series in order and every even position is filled with numbers from the Lucas sequence in order.
83. Input a matrix of size (m x n) and find the sum of the even elements and odd elements separately.
84. Input a square matrix of any order and find the sum of the elements of the two diagonals separately using an user defined function.
85. Input a matrix of size (m x n), transpose it and print the final transposed matrix.
86. Input two matrices of any order and perform addition, subtraction and multiplication on them.
87. Input a matrix of size (m x n) and interchange two of its rows according to the choice of the user.
88. Input a matrix of size (m x n) and interchange two of its columns according to the choice of the user.
89. Input a matrix of size (m x n) and check whether it is a sparse matrix or not. A sparse matrix is a matrix where the number of zero elements is greater than the number of non-zero elements.

PANDAS & DATA VISUALIZATION

90. Import the titanic dataset into a pandas dataframe and perform the following:
    i. List the column names.
    ii. List the first five rows for both columns 'survived' and 'age'.
    iii. Find the percentage of people who survived and who did not survive.
    iv. Find the percentage of women who survived.
    v. Find how many children below 5 years of age were on board the ship.
    vi. Find the number of children less than 5 years of age who survived.
    vii. How many embark points were there?
    viii. Find the average age of passengers for each class.
    ix. Draw a histogram to represent the number of survivors for each age group (0-10, 11-30, 31-60, >60).

Example:

```
import pylab as P
df['Age'].hist()
P.show()
```



x. Draw a histogram to represent the number of survivors for each sex.

xi. Find the correlation (if any) between passenger class and age.

xii. Draw a scatter plot between passenger age and fare and state your observations (if any).