Basic Python Program

1. Write a Python program to print "Hello Python".
2. Write a Python program to do arithmetical operations addition and division.
3. Write a Python program to find the area of a triangle.
4. Write a Python program to swap two variables.
5. Write a Python program to generate a random number.
6. Write a Python program to convert kilometres to miles.
7. Write a Python program to convert Celsius to Fahrenheit.
8. Write a Python program to display calendar.
9. Write a Python program to solve quadratic equation.
10. Write a Python program to swap two variables without temp variable.
11. Write a Python Program to Check if a Number is Positive, Negative or Zero.
12. Write a Python Program to Check if a Number is Odd or Even.
13. Write a Python Program to Check Leap Year.
14. Write a Python Program to Check Prime Number.
15. Write a Python Program to Print all Prime Numbers in an Interval of 1-10.
16. Write a Python Program to Find the Factorial of a Number.
17. Write a Python Program to Display the multiplication Table.
18. Write a Python Program to Print the Fibonacci sequence.
19. Write a Python Program to Check Armstrong Number?
20. Write a Python Program to Find Armstrong Number in an Interval.
21. Write a Python Program to Find the Sum of Natural Numbers.
22. Write a Python Program to Find LCM.
23. Write a Python Program to Find HCF.
24. Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal.
25. Write a Python Program to Find ASCII value of a character.
26. Write a Python Program to Make a Simple Calculator with 4 basic mathematical operations.
27. Write a Python Program to calculate your Body Mass Index
28. Write a Python Program to calculate the natural logarithm of any number
29. Write a Python Program for cube sum of first n natural numbers?
30. Write a Python Program to find sum of list elements.
31. Write a Python Program to find largest element in a list.
32. Write a Python Program to Split the list and add the first part to the end?
33. Write a Python Program to Sort Words in Alphabetic Order.

34. Write a Python Program to Remove Punctuation from a String.
35. Write a Python program to check if the given number is a Disarium Number.

    A Disarium number is a number that is equal to the sum of its digits each raised to the power of its respective position. For example, 89 is a Disarium number because (8^1) +(9^2) = 8 + 81 = 89.
36. Write a Python program to print all disarium numbers between 1 to 100.
37. Write a Python program to check if the given number is Happy Number.

    ***Happy Number**: A Happy Number is a positive integer that, when you repeatedly replace the number by the sum of the squares of its digits and continue the process, eventually reaches 1. If the process never reaches 1 but instead loops endlessly in a cycle, the number is not a Happy Number.*

    $1^2 + 9^2 = 82$

    $8^2 + 2^2 = 68$

    $6^2 + 8^2 = 100$

    $1^2 + 0^2 + 0^2 = 1$

    The process reaches 1, so 19 is a Happy Number.
38. Write a Python program to print all happy numbers between 1 and 100.
39. Write a Python program to determine whether the given number is a Harshad Number. A Harshad number (or Niven number) is an integer that is divisible by the sum of its digits. In other words, a number is considered a Harshad number if it can be evenly divided by the sum of its own digits. For example:

    18 is a Harshad number because 1+8 =9, and 18 is divisible by 9

    42 is not a Harshad number because 4+2 =6, and 42 is not divisible by 6.
40. Write a Python program to print all pronic numbers between 1 and 100.

    A pronic number, also known as an oblong number or rectangular number, is a type of figurate number that represents a rectangle. It is the product of two consecutive integers, n and (n + 1). Mathematically, a pronic number can be expressed as:

    $P_n = n * (n + 1)$

    For example, the first few pronic numbers are:

    - $P_1 = 1 * (1 + 1) = 2$
    - $P_2 = 2 * (2 + 1) = 6$
    - $P_3 = 3 * (3 + 1) = 12$
    - $P_4 = 4 * (4 + 1) = 20$

41. Write a Python program to Multiply all numbers in the list.
42. Write a Python program to find smallest number in a list.
43. Write a Python program to find largest number in a list.
44. Write a Python program to find second largest number in a list.
45. Write a Python program to find N largest elements from a list.
46. Write a Python program to print even numbers in a list.
47. Write a Python program to print odd numbers in a List.
48. Write a Python program to Remove Empty List from List.
49. Write a Python program to Cloning or Copying a list.
50. Write a Python program to Count occurrences of an element in a list.
51. Write a Python program to find words which are greater than given length k.
52. Write a Python program for removing i th character from a string.
53. Write a Python program to split and join a string.
54. Write a Python program to check if a given string is binary string or not.
55. Write a Python program to find uncommon words from two Strings.
56. Write a Python program to find all duplicate characters in string.
57. Write a Python Program to check if a string contains any special character. (use regex module)
58. Write a Python program to Extract Unique dictionary values.
59. Write a Python program to find the sum of all items in a dictionary.
60. Write a Python program to Merging two Dictionaries.
61. Write a Python program to convert key-values list to flat dictionary key_values_list = [('a', 1), ('b', 2), ('c', 3), ('d', 4)]
62. Write a Python program to sort Python Dictionaries by Key or Value.
63. Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.

    Suppose the following input is supplied to the program:
    without,hello,bag,world
    Then, the output should be: ***bag,hello,without,world***
64. Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically.
    Suppose the following input is supplied to the program: ***hello world and practice makes perfect and hello world again***

Then, the output should be: ***again and hello makes perfect practice world***

65. Write a program that accepts a sentence and calculate the number of letters and digits. Suppose the following input is supplied to the program:
    hello world! 123
    Then, the output should be:
    LETTERS 10
    DIGITS 3
66. A website requires the users to input username and password to register. Write a program to check the validity of password input by users.
    Following are the criteria for checking the password:
    1. At least 1 letter between [a-z]
    2. At least 1 number between [0-9] 1. At least 1 letter between [A-Z]
    3. At least 1 character from [$#@]
    4. Minimum length of transaction password: 6
    5. Maximum length of transaction password: 12
    Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma.
    Example If the following passwords are given as input to the program:
    ABd1234@1,a F1#,2w3E*,2We3345
    Then, the output of the program should be: **ABd1234@1**
67. Assuming that we have some email addresses in the "username@companyname.com (mailto:username@companyname.com)" format, please write program to print the user name of a given email address. Both user names and company names are composed of letters only.
    Example: If the following email address is given as input to the program: john@google.com (mailto:john@google.com) Then, the output of the program should be: john
68. Create a function that takes an angle in radians and returns the corresponding angle in degrees rounded to one decimal place. **Examples**
    radians_to_degrees(1) ⟶ 57.3
    radians_to_degrees(20) ⟶ 1145.9
    radians_to_degrees(50) ⟶ 2864.8
69. Given the side length x find the area of a hexagon. **Examples**
    area_of_hexagon(1) ⟶ 2.6
    area_of_hexagon(2) ⟶ 10.4

area_of_hexagon(3) ➞ 23.4

70. Create a function that replaces all the vowels in a string with a specified character. **Examples**

    replace_vowels("the aardvark", "#") ➞ "th# ##rdv#rk"

    replace_vowels("minnie mouse", "?") ➞ "m?nn?? m??s?"

    replace_vowels("shakespeare", "*") ➞ "sh*k*sp**r*"

71. Create a function that takes a list of non-negative integers and strings and return a new list without the strings. **Examples**

    filter_list([1, 2, "a", "b"]) ➞ [1, 2]

    filter_list([1, "a", "b", 0, 15]) ➞ [1, 0, 15]

    filter_list([1, 2, "aasf", "1", "123", 123]) ➞ [1, 2, 123]

72. The "Reverser" takes a string as input and returns that string in reverse order, with the opposite case. **Examples**

    reverse("Hello World") ➞ "DLROw OLLEh"

    reverse("ReVeRsE") ➞ "eSrEvEr"

    reverse("Radar") ➞ "RADAr"

73. Create a function that takes a single string as argument and returns an ordered list containing the indices of all capital letters in the string.

    Examples

    index_of_caps("eDaBiT") ➞ [1, 3, 5]

    index_of_caps("eQuINoX") ➞ [1, 3, 4, 6]

    index_of_caps("determine") ➞ []

    index_of_caps("STRIKE") ➞ [0, 1, 2, 3, 4, 5]

    index_of_caps("sUn") ➞ [1]

74. Create a function that takes the height and radius of a cone as arguments and returns the volume of the cone rounded to the nearest hundredth.

    Examples

    cone_volume(3, 2) ➞ 12.57

    cone_volume(15, 6) ➞ 565.49

    cone_volume(18, 0) ➞ 0

75. Create the function that takes a list of dictionaries and returns the sum of people's budgets.

Examples

get_budgets([

{ 'name': 'John', 'age': 21, 'budget': 23000 },

{ 'name': 'Steve', 'age': 32, 'budget': 40000 },

{ 'name': 'Martin', 'age': 16, 'budget': 2700 }

]) → 65700

get_budgets([

{ 'name': 'John', 'age': 21, 'budget': 29000 },

{ 'name': 'Steve', 'age': 32, 'budget': 32000 },

{ 'name': 'Martin', 'age': 16, 'budget': 1600 }

]) → 62600

76. Suppose that you invest $10,000 for 10 years at an interest rate of 6% compounded monthly. What will be the value of your investment at the end of the 10-year period? Create a function that accepts the principal p, the term in years t, the interest rate r, and the number of compounding periods per year n. The function returns the value at the end of term rounded to the nearest cent.

For the example:

compound_interest(10000, 10, 0.06, 12) → 18193.97

Note that the interest rate is given as a decimal and n=12 because with monthly compounding there are 12 periods per year. Compounding can also be done annually, quarterly, weekly, or daily.

Examples

compound_interest(100, 1, 0.05, 1) → 105.0

compound_interest(3500, 15, 0.1, 4) → 15399.26

compound_interest(100000, 20, 0.15, 365) → 2007316.26

77. Given a string of numbers separated by a comma and space, return the product of the numbers.

Examples

multiply_nums("2, 3") → 6

multiply_nums("1, 2, 3, 4") → 24

multiply_nums("54, 75, 453, 0") → 0

multiply_nums("10, -2") → -20

78.Create a function that squares every digit of a number

Examples

square_digits(9119) → 811181

square_digits(2483) → 416649

square_digits(3212) → 9414

Notes

The function receives an integer and must return an integer.

79.Create a function that returns the mean of all digits.

Examples

mean(42) → 3

mean(12345) → 3

mean(666) → 6

80.Write a function, that replaces all vowels in a string with a specified vowel.

Examples

vow_replace("apples and bananas", "u") → "upplus und bununus"

vow_replace("cheese casserole", "o") → "chooso cossorolo"

vow_replace("stuffed jalapeno poppers", "e") → "steffed jelepene peppers"