

Python SCRIPTS

PRACTICE DOCUMENT



July 1, 2019

Manali Gandhi

Table of Contents

[1. For loop to print a string 4](#_Toc12917268)

[2. If else to print numbers 4](#_Toc12917269)

[3. While loop for strings print 4](#_Toc12917270)

[4. Iterate over string in reverse using indexing 5](#_Toc12917271)

[5. Sum print, input numbers and sum print 5](#_Toc12917272)

[6. input string print, input and concat string print 5](#_Toc12917273)

[7. 3 variable program using nested if and else 6](#_Toc12917274)

[8. Ask user for name-if length is greater than 5 print it else type-not sufficient 6](#_Toc12917275)

[9. Reverse a string using string slicing and indexing 6](#_Toc12917276)

[10. To get the last part of a string using string slicing 7](#_Toc12917277)

[11. get user char, get user name, print the number of times the char appears in name 7](#_Toc12917278)

[12. User gives incorrcet input for character and name, put checks and print the number of times the char appears in name 7](#_Toc12917279)

[13. For loop for an empty string 8](#_Toc12917280)

[14. Define the magic number 8](#_Toc12917281)

[15. Get list in user input and print 9](#_Toc12917282)

[16. Using Loops to print elements of a list 9](#_Toc12917283)

[17. Use of len function in lists 10](#_Toc12917284)

[18. Use of range to print subset of list 10](#_Toc12917285)

[19. Use of max,min and sum, ord functions in list 11](#_Toc12917286)

[20. Use of chr function 12](#_Toc12917287)

[21. Sort the elements of a list of integers 13](#_Toc12917288)

[22. How to reverse a python list 13](#_Toc12917289)

[23. Sort the elements of a list of strings in alphabetical order 14](#_Toc12917290)

[24. Sort the elements of a list of strings in reverse alphabetical order 14](#_Toc12917291)

[25. Print an appended list and an extended list 15](#_Toc12917292)

[26. Print a Dictionary , use del keyword, clear,key,value,pop method and print 15](#_Toc12917293)

[27. Take user input to form a disctionary, print it 16](#_Toc12917294)

[28. Use try except for popitem for empty dict 17](#_Toc12917295)

[29. Use try except(KeyError) for printing the value of a dict key which doesn't exist 17](#_Toc12917296)

[30. Use get method in a dictionary 18](#_Toc12917297)

[31. Use Join function to convert a list into a string 19](#_Toc12917298)

[32. Use tuple() and set() to convert a list into tuple and set 19](#_Toc12917299)

[33. Use Zip() , iter() and len() function to convert a list into a dictionary 20](#_Toc12917300)

[34. List comprehension related programs 20](#_Toc12917301)

[35. Write a program using lambda() function 21](#_Toc12917302)

[36. Using count() method in lists 21](#_Toc12917303)

[37. Using counter method to convert lists to dictionary 21](#_Toc12917304)

[38. Split A Python List Into Evenly Sized Chunks using iter() and zip() functions 22](#_Toc12917305)

[39. Pyramid of stars, using 'end' keyword 23](#_Toc12917306)

[40. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old 23](#_Toc12917307)

[41. The Program when run should ask for the first name of day a person and on the correct name input, should print out the Birthday of the same person. And if the birthday is not found should print out the appropriate message 24](#_Toc12917308)

[42. Accept string from the user and display only those characters which are present at an even index 24](#_Toc12917309)

[43. Given two lists of integers; create a third list such that it should contain only odd numbers from the first list and even numbers from the second list 24](#_Toc12917310)

[44. Think of a Problem and code a solution for it. 25](#_Toc12917311)

[45. How to print the string :"yadoT si a doog yad" 26](#_Toc12917312)

[46. The Program when run should ask for the first name of a person and on the correct name input, should print out the Birthday of the same person. And if the birthday is not found ask it and update dictionary 26](#_Toc12917313)

[47. Store a list in a tuple and modify list 27](#_Toc12917314)

[48. More list comprehensions. 28](#_Toc12917315)

[49. Dictionary comprehension related programs 28](#_Toc12917316)

[50. Set/Tuple comprehension related programs 29](#_Toc12917317)

[51. Write a function that returns a string and use string operations in call itself 29](#_Toc12917318)

[52. Script to print number of lines in a paragraph 29](#_Toc12917319)

[53. Program to replace() ,format(), f string (use of replace and format functions and fstrings.)a string 30](#_Toc12917320)

[54. Write a function which accepts the default arguments 31](#_Toc12917321)

[55. Write a function which used arbitrary arguments as lists and directories and uses the same 31](#_Toc12917322)

[56. Pass a list and a directory to a arbitrary function and use those values 32](#_Toc12917323)

[57. Ask for a string from the user and tell him if it is a palindrome or not. Your program should work on numeric data as well 33](#_Toc12917324)

[58. Ask the user input to formulate a list of integers and print the output as a new list whose elements are the consecutive sums of itself and all the previous elements of the input list: 33](#_Toc12917325)

[59. Ask the user for two strings and let her know that the two inputs are anagrams or not: 34](#_Toc12917326)

[60. Ask the user for name and marks of 5 students and arrange them in order of the rankings 35](#_Toc12917327)

[61. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old 2](#_Toc13051176)

[62. The Program when run should ask for the first name of day a person and on the correct name input, should print out the Birthday of the same person. And if the birthday is not found should print out the appropriate message 2](#_Toc13051177)

[63. Accept string from the user and display only those characters which are present at an even index 3](#_Toc13051178)

[64. Given two lists of integers; create a third list such that it should contain only odd numbers from the first list and even numbers from the second list 3](#_Toc13051179)

[65. How to print the string :"yadoT si a doog yad" 4](#_Toc13051180)

[66. The Program when run should ask for the first name of a person and on the correct name input, should print out the Birthday of the same person. And if the birthday is not found ask it and update dictionary 4](#_Toc13051181)

[67. More list comprehensions. 5](#_Toc13051182)

[68. Write a function which accepts the default arguments 5](#_Toc13051183)

[69. Ask for a string from the user and tell him if it is a palindrome or not. Your program should work on numeric data as well 6](#_Toc13051184)

[70. Ask the user input to formulate a list of integers and print the output as a new list whose elements are the consecutive sums of itself and all the previous elements of the input list: 6](#_Toc13051185)

[71. Ask the user for two strings and let her know that the two inputs are anagrams or not: 7](#_Toc13051186)

[72. Ask the user for name and marks of 5 students and arrange them in order of the rankings 8](#_Toc13051187)

[73. Use the Input String: "Led 34215 Zeppelin 5620 is 384 the 143 greatest 789 band 876 ever"--> 10](#_Toc13051188)

1. For loop to print a string %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a="This is a string"

for i in a:

print (i)

b=input("Enter another string")

for i in b:

print (i)

For loop using slicing to print a string:

for i in range(0,15,3):

print(i)

sampleStr = "Hello!!"

print("\*\*\*\* Iterate over string using for loop\*\*\*\*")

for elem in sampleStr:

print(elem)

print("\*\*\*\* Iterate over string with index using range() \*\*\*\*")

for i in range( len(sampleStr) ):

print(sampleStr[i])

1. If else to print numbers %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=int(input("Enter number"))

if (a<10):

print (a)

else:

print ("big number")

1. While loop for strings print %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

coin=input("Enter heads or tails: ")

while (coin != "heads") and (coin != "tails"):

print ("Invalid choice")

coin=input("Enter heads or tails: ")

1. Iterate over string in reverse using indexing %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sampleStr=input("Enter a string to eb reversed: ")

i = len(sampleStr) - 1

while i >= 0 :

print(sampleStr[i])

i = i - 1

sampleStr=input("Enter a string to eb reversed: ")

i = 1

while i <= len(sampleStr) :

print(sampleStr[-i])

i = i + 1

1. Sum print, input numbers and sum print %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=int(input("Enter first number"))

b=int(input("Enter second number"))

c=a+b

print (c)

for i in range(a):

print ("We are",i,"and need to reach",a)

1. input string print, input and concat string print %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter first string")

b=input("Enter second string")

c=a+b

print (c)

for i in a:

print ("Your string is",a,"and other one is",b)

1. 3 variable program using nested if and else %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=int(input("Enetr num1: "))

b=int(input("Enter num2: "))

c=int(input("Enter num3: "))

if (a>b):

print ("From",a,"and",b,"bigger is",a)

if (a>c):

print (a,"is biggest")

else:

print (c,"is biggest")

else:

print ("From",a,"and",b,"bigger is",b)

if (b>c):

print (b,"is biggest")

else:

print (c,"is biggest")

1. Ask user for name-if length is greater than 5 print it else type-not sufficient %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter your name: ")

if (len(a)>5):

print ("Cool name")

else:

print ("Change your name")

1. Reverse a string using string slicing and indexing %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter your string: ")

i=1

while (i<=len(a)):

print (a[-i])

i=i+1

for j in range (1,len(a)+1):

print (a[-j])

print ("So the string reverse is", a[::-1])

1. To get the last part of a string using string slicing %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter your string: ")

b=len(a)-2

print ("slice string with second last characters is ", a[b:-1] )

print ("slice string with last 2 characters is ", a[b:len(a)+1] )

1. get user char, get user name, print the number of times the char appears in name %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter your name: ")

b=input("Enter favourite character: ")

j=0

for i in a:

if (i==b):

j=j+1

else:

continue

print ("Number of times",b,"appeared in your name is",j)

1. User gives incorrcet input for character and name, put checks and print the number of times the char appears in name %%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter your name: " )

b=input("Enter you favourite character: " )

c=input("Enter your agae: ")

while (a.isdigit() == True):

a=input("Your name enetered is not correct, please enter your name: " )

while (b.isdigit() == True):

b=input("Your character entered is not char, please enter a valid character " )

try:

val=int(c)

print("Your age is",c)

except ValueError:

print("You did not enter correct age")

j=0

for i in a:

if (b==i):

j=j+1

print(b,"occured",j,"times in your name",a)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter your name: 123

Enter you favourite character: 2

Enter your agae: we

Your name enetered is not correct, please enter your name: Manali

Your character entered is not char, please enter a valid character a

You did not enter correct age

a occured 2 times in your name Manali

1. For loop for an empty string %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter a string: ")

b=a.strip()

if not b:

print("the string you enetered is empty, please enter a valid string: ")

else:

print("string is good , you may procedd")

1. Define the magic number

Take input from user as a number: Ask him to try to guess, If matches : Congratulation, If greater : Tell him it's greater, If smaller : Tell him it's smaller, Do this until he guesses the correct number. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=int(input("Guess the magic number: "))

b=17

while (a != b):

if (a>b):

a=int(input("The number entered is more than Magic number, try again: "))

else:

a=int(input("The number entered is less than Magic number, try again: "))

print("You guessed it right, Congratulations!!!!")

1. Get list in user input and print %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter a list of elemnets separated by space: ")

list=a.split()

print (list)

b=input("Enter a list separated by comma: ")

list1=b.split(",")

print (list1)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter a list of elemnets separated by space: 1 2 3 4 5

['1', '2', '3', '4', '5']

Enter a list separated by comma: 1,2,2,3,4,5,6,7,8,9

['1', '2', '2', '3', '4', '5', '6', '7', '8', '9']

1. Using Loops to print elements of a list %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter the elements of a list separated by @: ")

list=a.split("@")

j=0

for i in list:

print ("The element of the list is",list[j])

print ("The value of i is",i)

j=j+1

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the elements of a list separated by @: a@b2c@d@e@f

The element of the list is a

The value of i is a

The element of the list is b2c

The value of i is b2c

The element of the list is d

The value of i is d

The element of the list is e

The value of i is e

The element of the list is f

The value of i is f

1. Use of len function in lists %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter the elments of a list separated by space: ")

list=a.split()

length=len(list)

i=0

while (i<length):

print ("Length of list is ",length)

print ("Current element of list is ",list[i])

i=i+1

a=input("Enter the elments of a list separated by space: ")

list=a.split()

i=0

while (i<len(list)):

print ("Length of list is ",len(list))

print ("Current element of list is ",list[i])

i=i+1

1. Use of range to print subset of list %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=list([1,2,3,4,5,6,7,8,9])

b=int(input("Enter the limit of list to print: "))

for i in range(0,b):

print ("the current element is ",a[i])

1. Use of max,min and sum, ord functions in list %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter the elements of a list separated by space: ")

list=a.split()

print ("maximun value in the list is: ",max(list))

print ("minimum value in the list is: ",min(list))

print ("Sum of elements of a list is: ",sum(list))

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the elements of a list separated by space: 1 3 5 10 2 4 6 a b r abc

maximun value in the list is: r

minimum value in the list is: 1

Traceback (most recent call last):

File ".\temp.py", line 5, in <module>

print ("Sum of elements of a list is: ",sum(list))

TypeError: unsupported operand type(s) for +: 'int' and 'str'

[=91

A=65

a=97

a=input("Enter the elements of a list separated by space: ")

list=a.split()

print ("maximun value in the list is: ",max(list))

print ("minimum value in the list is: ",min(list))

#print ("Sum of elements of a list is: ",sum(list))

for i in range(len(list)):

try:

val=int(list[i])

print ("The element at index",i,"is a integer",list[i])

except ValueError:

print ("The element at index",i,"is a string with ASCII value",ord(list[i]))

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the elements of a list separated by space: a 1 b 3 A 10 b 12

maximun value in the list is: b

minimum value in the list is: 1

The element at index 0 is a string with ASCII value 97

The element at index 1 is a integer 1

The element at index 2 is a string with ASCII value 98

The element at index 3 is a integer 3

The element at index 4 is a string with ASCII value 65

The element at index 5 is a integer 10

The element at index 6 is a string with ASCII value 98

The element at index 7 is a integer 12

1. Use of chr function %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i in range(65,97,1):

print ("the charcter for the ASCII value of",i,"is ",chr(i))

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

the charcter for the ASCII value of 65 is A

the charcter for the ASCII value of 66 is B

the charcter for the ASCII value of 67 is C

the charcter for the ASCII value of 68 is D

the charcter for the ASCII value of 69 is E

the charcter for the ASCII value of 70 is F

the charcter for the ASCII value of 71 is G

the charcter for the ASCII value of 72 is H

the charcter for the ASCII value of 73 is I

the charcter for the ASCII value of 74 is J

the charcter for the ASCII value of 75 is K

the charcter for the ASCII value of 76 is L

the charcter for the ASCII value of 77 is M

the charcter for the ASCII value of 78 is N

the charcter for the ASCII value of 79 is O

the charcter for the ASCII value of 80 is P

the charcter for the ASCII value of 81 is Q

the charcter for the ASCII value of 82 is R

the charcter for the ASCII value of 83 is S

the charcter for the ASCII value of 84 is T

the charcter for the ASCII value of 85 is U

the charcter for the ASCII value of 86 is V

the charcter for the ASCII value of 87 is W

the charcter for the ASCII value of 88 is X

the charcter for the ASCII value of 89 is Y

the charcter for the ASCII value of 90 is Z

the charcter for the ASCII value of 91 is [

the charcter for the ASCII value of 92 is \

the charcter for the ASCII value of 93 is ]

the charcter for the ASCII value of 94 is ^

the charcter for the ASCII value of 95 is \_

the charcter for the ASCII value of 96 is `

1. Sort the elements of a list of integers %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Works:-

a=input("Enter the elements of a list separated by space: ")

lista=a.split()

print("unsorted list is",lista)

lista.sort()

listb=lista

print("Sorted list is",listb)

Question - Doesn't work:-

a=input("Enter the elements of a list separated by space: ")

lista=a.split()

print("unsorted list is",lista)

listb=lista.sort()

print("Sorted list is",listb)

1. How to reverse a python list %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Works:-

a=input("Enter the elements of a list separated by space: ")

lista=a.split()

print("Original list is",lista)

lista.reverse()

print("Reversed list is",lista)

Question - Doesn't work:-

a=input("Enter the elements of a list separated by space: ")

lista=a.split()

print("Original list is",lista)

print("Reversed list is",lista.reverse())

1. Sort the elements of a list of strings in alphabetical order %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter a list of strings separated by a comma: ")

list=a.split(',')

print("Original list is",list)

list.sort()

print("Alphabetized list is",list)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter a list of strings separated by a comma: manali,shekhar,Shekhar,Manali,Manal

Original list is ['manali', 'shekhar', 'Shekhar', 'Manali', 'Manal']

Alphabetized list is ['Manal', 'Manali', 'Shekhar', 'manali', 'shekhar']

1. Sort the elements of a list of strings in reverse alphabetical order %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter a list of strings separated by a comma: ")

list=a.split(',')

print("Original list is",list)

list.sort()

list.reverse()

print("Reverse alphabetized list is",list)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter a list of strings separated by a comma: manali,Mana,Manali,shekhar,Shek,SHEKHAR

Original list is ['manali', 'Mana', 'Manali', 'shekhar', 'Shek', 'SHEKHAR']

Reverse alphabetized list is ['shekhar', 'manali', 'Shek', 'SHEKHAR', 'Manali', 'Mana']

1. Print an appended list and an extended list %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter a list separated by space: ")

lista=a.split()

print("You list is",a.split())

b=input("you want to enter a new list as an element of teh old list, enter it: ")

listb=b.split()

print("New list elment to be added is",listb)

lista.append(listb)

print("Appended list is",lista)

listc=a.split()

listc.extend(listb)

print("Extended list is",listc)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter a list separated by space: 1 2 3

You list is ['1', '2', '3']

you want to enter a new list as an element of teh old list, enter it: 4 5

New list elment to be added is ['4', '5']

Appended list is ['1', '2', '3', ['4', '5']]

Extended list is ['1', '2', '3', '4', '5']

1. Print a Dictionary , use del keyword, clear,key,value,pop method and print outputs%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter the input values to the dictionary separated by comma: ")

my\_dict={}

lista=a.split(',')

i=0

j=1

k=int(len(lista)/2)

for dict in range(1,k+1):

my\_dict[lista[i]]=lista[j]

i=i+2

j=j+2

print("The dictionary is ",my\_dict)

b=input("Enter the element t be deleted")

del my\_dict[b]

print("Dictionary after deleting element",b,"is",my\_dict)

print("The keys are",my\_dict.keys())

print("The values are",my\_dict.values())

print("The items are",my\_dict.items())

c=input("Enter the key for element needs to be removed: ")

my\_dict.pop(c)

print("Dictionary after popping teh element is",my\_dict)

my\_dict.clear()

print("clearing teh dictionary, your dictionary is now:",my\_dict)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the input values to the dictionary separated by comma: 1,2,3,4,5,6,7,8

The dictionary is {'1': '2', '3': '4', '5': '6', '7': '8'}

Enter the element t be deleted3

Dictionary after deleting element 3 is {'1': '2', '5': '6', '7': '8'}

The keys are dict\_keys(['1', '5', '7'])

The values are dict\_values(['2', '6', '8'])

The items are dict\_items([('1', '2'), ('5', '6'), ('7', '8')])

Enter the key for element needs to be removed: 5

Dictionary after popping teh element is {'1': '2', '7': '8'}

clearing teh dictionary, your dictionary is now: {}

1. Take user input to form a disctionary, print it %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter the key, values of the dictionary separated by comma: ")

my\_dict={}

i=0

j=1

lista=a.split(',')

k=int(len(lista)/2)

print("k is",k)

for dict in range(1,k+1):

my\_dict[lista[i]]=lista[j]

i=i+2

j=j+2

print("Your dictionary is",my\_dict)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the key, values of the dictionary separated by comma: 1,2,3,4,5,6

k is 3

Your dictionary is {'1': '2', '3': '4', '5': '6'}

1. Use try except for popitem for empty dict %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter the elements of dictionary separated by spaces:")

lista=a.split()

my\_dict={}

i=0

j=1

k=int(len(lista)/2)

for dict in range(1,k+1):

my\_dict[lista[i]]=lista[j]

i=i+2

j=j+2

print("The dictionary is: ",my\_dict)

try:

my\_dict.popitem()

print("Your dictionary after popitem is now", my\_dict)

except KeyError:

print("Your dictionary is empty", my\_dict)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the elements of dictionary separated by spaces:1 2 3 4

The dictionary is: {'1': '2', '3': '4'}

Your dictionary after popitem is now {'1': '2'}

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the elements of dictionary separated by spaces:

The dictionary is: {}

Your dictionary is empty {}

1. Use try except(KeyError) for printing the value of a dict key which doesn't exist %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Eneter the elements of dictionary separated by space: ")

lista=a.split()

my\_dict={}

i=0

j=1

k=int(len(lista)/2)

for dict in range(1,k+1):

my\_dict[lista[i]]=lista[j]

i=i+2

j=j+2

print("Your dictionay is:",my\_dict)

b=input("Enter the key for which you want to see teh value: ")

try:

print("The value for key",b,"is",my\_dict[b])

except KeyError:

print("The key you entered does not exist")

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Eneter the elements of dictionary separated by space: 1 2 3 4

Your dictionay is: {'1': '2', '3': '4'}

Enter the key for which you want to see teh value: 5

The key you entered does not exist

1. Use get method in a dictionary %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter the elements of dictionary separated by spaces:")

lista=a.split()

my\_dict={}

i=0

j=1

k=int(len(lista)/2)

for dict in range(1,k+1):

my\_dict[lista[i]]=lista[j]

i=i+2

j=j+2

print("The dictionary is: ",my\_dict)

b=input("Enter the key for whic you want to see teh value: ")

x=my\_dict.get(b)

print("The value for key",b, "is",x)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the elements of dictionary separated by spaces:1 2 3 4 5 6

The dictionary is: {'1': '2', '3': '4', '5': '6'}

Enter the key for whic you want to see teh value: 3

The value for key 3 is 4

1. Use Join function to convert a list into a string %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter the elements of dictionary separated by spaces:")

lista=a.split()

print("The list is: ",lista)

b=''.join(lista)

print("The string formed from list elements is: ",b)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the elements of dictionary separated by spaces:1 2 3

The list is: ['1', '2', '3']

The string formed from list elements is: 123

1. Use tuple() and set() to convert a list into tuple and set %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter the elements of dictionary separated by spaces:")

lista=a.split()

print("The list is: ",lista)

print("The tuple formed is: ",tuple(lista))

print("The set formed is: ",set(lista))

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the elements of dictionary separated by spaces:1 2 3 4

The list is: ['1', '2', '3', '4']

The tuple formed is: ('1', '2', '3', '4')

The set formed is: {'3', '2', '4', '1'}

# Use Zip() , iter() and len() function to convert a list into a dictionary

%%%%%%%%%%%%%%%%% Read further about Zip-didn't get--------------

a = [1, 2, 3, 4, 5, 6]

print("Length of list a is: ",len(a))

# Create a list iterator object

i = iter(a)

# Zip and create a dictionary

print(dict(zip(i, i)))

>>> a = dict(one=1, two=2, three=3)

>>> b = {'one': 1, 'two': 2, 'three': 3}

>>> c = dict(zip(['one', 'two', 'three'], [1, 2, 3]))

>>> d = dict([('two', 2), ('one', 1), ('three', 3)])

>>> e = dict({'three': 3, 'one': 1, 'two': 2})

>>> a == b == c == d == e

True

1. List comprehension related programs %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[x\*\*2 for x in range(10)]

Out[1]: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

[x\*\*2 for x in range(10) if x%2==0]

Out[1]: [0, 4, 16, 36, 64]

using list comprehension to transform your lists into other lists:

myList = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[(lambda x: x\*x)(x) for x in myList]

Out[1]: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

P.S.: lambda functions are anonymous functions that are constructed at runtime. Considering that that piece of code can just serve as an indication for you, you can just ignore it and focus on x\*x. You see that this function takes one element and multiplies it by itself. It’s an anonymous function, so you need to pass it your list element to make sure that the lambda function takes that one as input. That is why you see (lambda x: x\*x)(x)

1. Write a program using lambda() function %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

list1=[1,2,3,4]

f=lambda x : x\*3

list2=[f(x) for x in list1]

print (list2)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

[3, 6, 9, 12]

1. Using count() method in lists %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter elements of a list separated by spaces: ")

b=input("Eneter teh element you want to count: ")

list1=a.split()

print ("Occurence of",b,"is",list1.count(b))

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter elements of a list separated by spaces: 2 3 4 4 4 5 6 7 3

Eneter teh element you want to count: 3

Occurence of 3 is 2

1. Using counter method to convert lists to dictionary %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

# Import `Counter` from the `collections` library

from collections import Counter

list = ["a","b","b"]

# Pass `list` to `Counter()`

print(Counter(list))

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Counter({'b': 2, 'a': 1})

# Split A Python List Into Evenly Sized Chunks using iter() and zip() functions

%%%%%%%%%%%%% Read more about Zip and iter functions---------

# Your list `x`

x = [1,2,3,4,5,6,7,8,9]

# Split `x` up in chunks of 3

y = zip(\*[iter(x)]\*3)

# Use `list()` to print the result of `zip()`

list(y)

Out[1]: [(1, 2, 3), (4, 5, 6), (7, 8, 9)]

The code above works as follows:

iter() is an iterator over a sequence.

[iter(x)] \* 3 produces a list with three listiterator objects: each list iterator is an iterator of x.

the \* that is passed to the zip() function before anything else unpacks the sequence into arguments so that you’re passing the same iterator three times to the zip() function, and it pulls an item from the iterator each time.

Wait. That last step is totally not clear.

Let me guide you step by step:

You will have three list iterator objects, which you can think of as:

[1,2,3,4,5,6,7,8,9], [1,2,3,4,5,6,7,8,9], [1,2,3,4,5,6,7,8,9]

The first time, zip() will take one element of the list sequentially, which leaves you with:

[1][2][3]

Note that the iterator objects will keep track for you which element is next in the iteration!

The second time, elements will be added to the three lists you just created, and you will end up with:

[1, 4], [2, 5], [3, 6]

The last time, you follow the same procedure, and you end up with:

[1, 4, 7], [2, 5, 8], [3, 6, 9]

Zipping these three lists together will leave you with:

[1, 2, 3], [4, 5, 6], [7, 8, 9]

If, however, you decide to put this logic into a function, you can follow this approach:

# Method to split up your lists into chunks

def chunks(list, chunkSize):

"""Yield successive chunkSize-sized chunks from list."""

for i in range(0, len(list), chunkSize):

yield list[i:i + chunkSize]

# Use your `chunks` function to print out chunks of the same size

import pprint

pprint.pprint(list(chunks(range(10, 75), 10)))

1. Pyramid of stars, using 'end' keyword %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

The end=' ' is just to say that you want a space after the end of the statement instead of a new line character. In Python 2.x you would have to do this by placing a comma at the end of the print statement. The end parameter means that the line gets ' ' at the end rather than a newline character.

a=int(input("Enter the number of rows: "))

for i in range(1,a+1):

for j in range(1,i+1):

print ("\*",end=" ")

print() ----------------------------------> To print teh next iteration in next line

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the number of rows: 5

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

1. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

import datetime

a=input("Enter your name and age separated by space: ")

lista=a.split()

b=100-int(lista[1])

now = datetime.datetime.now() ----------------------------------------> Check how we are accessing teh datetime.now function

year\_of\_100=int(now.year)+b

print ("Hi", lista[0],"your age will be 100 years in", year\_of\_100)

print (dir(datetime.datetime))

1. The Program when run should ask for the first name of day a person and on the correct name input, should print out the Birthday of the same person. And if the birthday is not found should print out the appropriate message %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

my\_dict={'Manali':'17may','Sonali':'13oct','Shekhar':'22sep','Shivam':'5oct'}

a=input("Enter teh name who's bday yu wish to see: ")

try:

print ("The Birthday of",a,"is",my\_dict[a])

except keyerror:

print ("This person's bday is not known")

Method2: ---------------------------------------------------------------------------> Use of format function

birthday={'Manali':'17may','Sonali':'13oct','Shekhar':'22sep','Shivam':'5oct'}

name=input("Enter teh name who's bday yu wish to see: ")

if name in birthday.keys():

print("{}'s birthday is on {}".format(name,my\_dict[name])

else:

print("Sorry birthday not known for {}".format(name))

with Format function you don't worry about the cast an int as a string while printing.

# Accept string from the user and display only those characters which are present at an even index

%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter teh string: ")

b=len(a)

print("The string even chars are: ",a[0:b:2])

1. Given two lists of integers; create a third list such that it should contain only odd numbers from the first list and even numbers from the second list %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=[1,2,3,4,5,6]

b=[7,8,9,10,11,12]

j=0

k=0

lista=[]

for i in a:

if (i%2==0):

lista.append(i)

j=j+1

else:

continue

for i in b:

if (i%2==0):

continue

else:

lista.append(i)

j=j+1

print ("The final list is: ",lista)

output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

The final list is: [2, 4, 6, 7, 9, 11]

1. Think of a Problem and code a solution for it. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Take it forward and implement it

Reminding people about weekly status completion:

import smtplib

import datetime

# creates SMTP session

s = smtplib.SMTP('smtp.gmail.com', 587)

# start TLS for security

s.starttls()

# message to be sent

message="Please complete your weekly status"

# Authentication

s.login("manali.gslab@gmail.com", "Inspiron@1")

if b == 0:

# sending the mail

s.sendmail("manali.gslab@gmail.com","manali.gandhi@gslab.com",message)

# terminating the session

s.quit()

else:

print ("Not the right day")

Other important links :-

https://stackoverflow.com/questions/4536146/need-an-python-script-that-uses-skype4py-to-send-an-instant-message

https://sevabot-skype-bot.readthedocs.io/en/latest/python.html

1. How to print the string :"yadoT si a doog yad" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Method1:

a=input("Enter a sentence: ")

list1=a.split()

for i in list1:

print (i[::-1],end=' ')

b=''.join(list1)

Method2:

a=input("Enter a sentence: ")

list1=a.split()

list2=[]

for i in list1:

i=i[::-1]

list2.append(i)

b=' '.join(list2)

print (b)

1. The Program when run should ask for the first name of a person and on the correct name input, should print out the Birthday of the same person. And if the birthday is not found ask it and update dictionary %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Method1:

birthday={'Manali':'17may','Sonali':'13oct','Shekhar':'22sep','Shivam':'5oct'}

name=input("Enter the name who's bday yu wish to see: ")

if name in birthday.keys():

print("{}'s birthday is on {}".format(name,birthday[name]))

else:

b=input("Sorry birthday not known ,Please enter the birthday: ")

birthday[name]=b

print ("Final dictionary is:",birthday )

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the name who's bday yu wish to see: Shimi

Sorry birthday not known ,Please enter the birthday: 5oct

Final dictionary is: {'Manali': '17may', 'Sonali': '13oct', 'Shekhar': '22sep', 'Shivam': '5oct', 'Shimi': '5oct'}

1. Store a list in a tuple and modify list %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#a=input("Enter the elements of the list separate by space: ")

#tup1=tuple(a.split())

tup1=('1', '2', '3', '4',[5,6,7])

print (tup1)

tup1[4][1]=7

print (tup1)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

('1', '2', '3', '4', [5, 6, 7])

('1', '2', '3', '4', [5, 7, 7])

1. All the numbers in range 1-100 both included, write a list that is perfect squares. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=[x\*x for x in range(12) if x\*x <= 100]

print (a)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

# More list comprehensions.

More List comprehension related programs %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

list= ["a","b","b"]

[[x,list.count(x)] for x in set(list)]

Out[1]: [['a', 1], ['b', 2]]

list= ["a","b","b"]

[[x,list.count(x)] for x in list]

Out[1]: [['a', 1], ['b', 2], ['b', 2]]

print ([x\*x for x in range(1,10)])

[x\*\*2 for x in range(10)]

Out[1]: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

[x\*\*2 for x in range(10) if x%2==0]

Out[1]: [0, 4, 16, 36, 64]

using list comprehension to transform your lists into other lists:

myList = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[(lambda x: x\*x)(x) for x in myList]

Out[1]: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

a=[1,2,3,4,5,6,7,8,9]

list=[x for x in a]

list=[x\*x for x in a]

list=[x for x in a if x%2==0]

list=[x\*x for x in a if x%2==0]

list=[x\*x for x in a if x%2!=0]

a=[x\*x for x in range(12) if x\*x <= 100]

\*\*\*\*\*\*\*\* letter number combo list

string\_letter='abcd'

list\_comb\_direct=[(letter, num) for letter in string\_letter for num in range(4)]

print(list\_comb\_direct)

\*\*\*\*\*\*\*\*\* Listing all prime numbers in a range

def prime\_check(random\_number):

for i in range(2, random\_number):

if random\_number % i == 0:

return False

break

else:

continue

return True

prime\_range = int(input("Enter the range of prime numbers you want to list: "))

list\_all = [x for x in range(2, prime\_range+1)] -----------------------> +1 will include numbers including the number you give.

list\_prime\_range = list(filter(prime\_check, list\_all))

print(list\_all)

print(list\_prime\_range)

\*\*\*\*\*\*\*\*

list\_comp = [x for x in range(0, 100) if x % 15 == 0]

print(list\_comp)

C:\Users\GS-0838\AppData\Local\Programs\Python\Python37\python.exe C:/Users/GS-0838/Desktop/Roadmap/Python/temp.py

[0, 15, 30, 45, 60, 75, 90]

1. Dictionary comprehension related programs %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

<http://cmdlinetips.com/2018/01/5-examples-using-dict-comprehension/>

<https://www.geeksforgeeks.org/python-dictionary-comprehension/>

list1 = ['A', 'B', 'C', 'D', 'E', 'F']

list2 = ['a', 'b', 'c', 'd', 'e', 'f']

my\_dict = {x: y for x, y in zip(list1, list2)}

print (my\_dict)

Output:

C:\Users\GS-0838\AppData\Local\Programs\Python\Python37\python.exe C:/Users/GS-0838/Desktop/Roadmap/Python/temp.py

{'A': 'a', 'B': 'b', 'C': 'c', 'D': 'd', 'E': 'e', 'F': 'f'}

\*\*\*\*\*\*\*\*

list1 = ['A', 'B', 'C', 'D', 'E', 'F']

list2 = ['a', 'b', 'c', 'd', 'e', 'f']

list3 = [1, 2, 3, 4, 5, 6]

my\_dict = {x: (y,z) for x,y,z in zip(list1, list2, list3)}

print (my\_dict)

Output:

C:\Users\GS-0838\AppData\Local\Programs\Python\Python37\python.exe C:/Users/GS-0838/Desktop/Roadmap/Python/temp.py

{'A': ('a', 1), 'B': ('b', 2), 'C': ('c', 3), 'D': ('d', 4), 'E': ('e', 5), 'F': ('f', 6)}

\*\*\*\*\*\*\*\* skipping elements using if

list1 = [1, 2, 3, 4, 5]

list2 = ['a', 'b', 'c', 'd', 'e', 'f', 'g']

my\_dict = {x: y for x,y in zip(list1, list2) if x != 3}

print(my\_dict)

Output:

C:\Users\GS-0838\AppData\Local\Programs\Python\Python37\python.exe C:/Users/GS-0838/Desktop/Roadmap/Python/temp.py

{1: 'a', 2: 'b', 4: 'd', 5: 'e'}

1. Set/Tuple comprehension related programs %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

list1 =[1, 2, 3, 1, 2, 2, 4, 5, 6, 7, 3, 7, 8, 9, 10, 11, 11, 23, 22, 12, 22]

my\_set={x for x in list1}

print(my\_set)

\*\*\*\*\*\*\*

myList = [1, 2, 3, 4, 4, 5, 6, 6, 7, 8, 9, 9, 10]

b={(lambda x: x\*x)(x) for x in myList}

print (b)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

{1, 4, 9, 16, 25, 36, 49, 64, 81, 100}

# Generator Expressions

Generator functions don't hold entire result in memory, they yield the result at runtime.---------------> yield() and next() functions

Hence it is better with performance.

def square\_nums(nums):

for i in nums:

yield (i\*i)

my\_num = square\_nums([1, 2, 3, 4, 5])

print(my\_num)

print(next(my\_num))

print(next(my\_num))

print(next(my\_num))

print(next(my\_num))

print(next(my\_num))

Output:

C:\Users\GS-0838\AppData\Local\Programs\Python\Python37\python.exe C:/Users/GS-0838/Desktop/Roadmap/Python/temp.py

<generator object square\_nums at 0x000002970A916C00>

1

4

9

16

25

\*\*\*\*\*\*\*

def square\_nums(nums):

for i in nums:

yield (i\*i)

my\_num = square\_nums([1, 2, 3, 4, 5])

for i in my\_num:

print(i)

Output:

C:\Users\GS-0838\AppData\Local\Programs\Python\Python37\python.exe C:/Users/GS-0838/Desktop/Roadmap/Python/temp.py

1

4

9

16

25

\*\*\*\*\*\* Generator function in the form of generator comprehensions

square\_nums = (x\*x for x in [1, 2, 3, 4, 5])

print(square\_nums)

for i in square\_nums:

print(i)

Output:

C:\Users\GS-0838\AppData\Local\Programs\Python\Python37\python.exe C:/Users/GS-0838/Desktop/Roadmap/Python/temp.py

<generator object <genexpr> at 0x00000272E3806C00>

1

4

9

16

25

# Write a function that returns a string and use string operations in call itself

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

def func():

return 'Helllo'

print (func().upper())

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

HELLLO

1. Script to print number of lines in a paragraph %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=[1,2,3,4,5,6,7,8,9]

list=[x\*x for x in a if x%2!=0]

print ("""This is paragraph to be printed.

Python is a user friendly scripting language.

We use python along with Robot in our automation framework.

It is easy to analyse and debug.""",list)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

This is paragraph to be printed.

Python is a user friendly scripting language.

We use python along with Robot in our automation framework.

It is easy to analyse and debug. [1, 9, 25, 49, 81]

1. Program to replace() ,format(), f string (use of replace and format functions and fstrings.)a string %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter the string : ")

b=input("Enter the substring that needs to be replaced: ")

c=input("Enter the substring that you need to replace it with: ")

replaced\_string=a.replace(b,c)

print ("Original string is",a)

print ("Replaced string is",replaced\_string)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the string : Good morning

Enter the substring that needs to be replaced: morning

Enter the substring that you need to replace it with: night

Original string is Good morning

Replaced string is Good night

a=input("Enter the string : ")

b=input("Enter the substring that needs to be replaced: ")

c=input("Enter the substring that you need to replace it with: ")

replaced\_string=a.replace(b,c)

message=' "{}" is the original string, "{}" is the final string'.format(a,replaced\_string)

print ('{} is the original string, and {} is the replaced string'.format(a,replaced\_string))

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the string : Good morning

Enter the substring that needs to be replaced: morning

Enter the substring that you need to replace it with: night

"Good morning" is the original string, "Good night" is the final string

Good morning is the original string, and Good night is the replaced string

a=input("Enter the string : ")

b=input("Enter the substring that needs to be replaced: ")

c=input("Enter the substring that you need to replace it with: ")

replaced\_string=a.replace(b,c)

message=f' "{a}" is the original string, "{replaced\_string}" is the final string'

print (message)

print (f' {a} is the original string, {replaced\_string} is the final string')

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the string : Good morning

Enter the substring that needs to be replaced: morning

Enter the substring that you need to replace it with: evening

"Good morning" is the original string, "Good evening" is the final string

Good morning is the original string, Good evening is the final string

https://www.youtube.com/watch?v=9Os0o3wzS\_I&list=PL-osiE80TeTskrapNbzXhwoFUiLCjGgY7&index=8

1. Write a function which accepts the default arguments %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

def greeting(name,greet = 'day'): -----------------------------------> Use of format function and f string

statement=f' Hi {name}!, Good {greet} '

# statement=' Hi {}!, Good {} '.format(name,greet)

print (statement)

greeting('Manali','morning')

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Hi Manali!, Good morning

1. Write a function which used arbitrary arguments as lists and directories and uses the same %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

-------------------------------------------------------------------> \*args, output is tuple.

print(args)

def arbitrary(\*args):

arbitrary('a','b','c')

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

('a', 'b', 'c')

-------------------------------------------------------------------> \*args, \*\*kwargs, output is tuple and dictionary

def arbitrary(\*args,\*\*kwargs):

print(args)

print(kwargs)

arbitrary('Math','English',name='John',age=25)

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

('Math', 'English')

{'name': 'John', 'age': 25}

1. Pass a list and a directory to a arbitrary function and use those values %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

-------------------------------------------------------------------> passing a list and dictionary both to \*args

def arbitrary(\*args):

print(args)

a=[1,2,3,4]

b={'Name':'Manali','age':32}

arbitrary(a,b)

output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

([1, 2, 3, 4], {'Name': 'Manali', 'age': 32})

-------------------------------------------------------------------> passing a list and dictionary to \*args and \*\*kwargs

Method1:

def arbitrary(\*args,\*\*kwargs):

print(args)

print(kwargs)

a=[1,2,3,4]

b={'Name':'Manali','age':32}

arbitrary(a,b)

output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

([1, 2, 3, 4], {'Name': 'Manali', 'age': 32})

{}

Method2: -------------------------------------------------------------------> Defining unpacking for list and dictionary separately.

def arbitrary(\*args,\*\*kwargs):

print(args)

print(kwargs)

a=[1,2,3,4]

b={'Name':'Manali','age':32}

arbitrary(\*a,\*\*b)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

(1, 2, 3, 4)

{'Name': 'Manali', 'age': 32}

# Ask for a string from the user and tell him if it is a palindrome or not. Your program should work on numeric data as well

Examples of a Palindrome: "level", "madam", 1234321 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

def palindrome(a):

if a[::-1]==a:

return (True)

else:

return (False)

my\_str=input("Enter a string you want to check for palindrome: ")

if palindrome(my\_str)== True:

print("String is palindrome")

else:

print("Sring is not palindrome")

1. Ask the user input to formulate a list of integers and print the output as a new list whose elements are the consecutive sums of itself and all the previous elements of the input list: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#Example Input : [1,2,3,4,5] O/P: [1,3,6,10,15]

#a[0]=a[0]

#a[1]=a[1]+a[0]

#a[2]=a[2]+a[1]

#a[n]=a[n]+a[n-1]

--------------------------------------------------------------------------> Use of list extensions, list passing, tuple to list conversion.

def my\_sum(\*nums):

nums\_new=[int(x) for x in nums]

for i in range(len(nums\_new)-1):

nums\_new[i+1]=nums\_new[i+1]+nums\_new[i]

return (nums\_new)

list=input("Enter the list of numbers to be added separated by space: ")

num\_list=list.split()

result\_list=my\_sum(\*num\_list)

print(f'The original list is {num\_list}, final list is {result\_list}')

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the list of numbers to be added separated by space: 1 2 3

The original list is ['1', '2', '3'], final list is [1, 3, 6]

1. Ask the user for two strings and let her know that the two inputs are anagrams or not: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Example of an anagram: LISTEN and SILENT

def anagram(a,b):

set1=set([i for i in a])

set2=set([i for i in b])

if set1 == set2:

return (True)

else:

return (False)

str\_input=input("Enter two strings to be checked for anagram separaed by comma: ")

str1,str2=str\_input.split(',')

if anagram(str1,str2)== True:

print("The strings entered are anagram")

else:

print("The strings entered are not anagram")

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter two strings to be checked for anagram separaed by comma: LISTEN,SILENT

The strings entered are anagram

# Ask the user for name and marks of 5 students and arrange them in order of the rankings

i.e. the one with the highest marks gets the Rank 1, the second highest, Rank2 and so on. In case there is a tie between two students the Name in alphabetical order is the tiebreaker. For examlple, if Ram and Rohit have 80 marks each then Ram gets a rank higher than that of Rohit's %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\*\* Check out bubble sort---------------------------------------> Implement this for same number of marks and name starting with same alphabet.

#Ask the user for name and marks of 5 students and arrange them in order of the rankings i.e. the one with the highest marks gets the Rank 1, the second highest, Rank2 and so on. In case there is a tie between two students the Name in alphabetical order is the tiebreaker. For examlple, if Ram and Rohit have 80 marks each then Ram gets a rank higher than that of Rohit's

#RAM:80 , Rohit:90 , Manali:100, Sonali:85, Shekhar:90

#RAM,80,Rohit,90,Manali,70,Sonali,85,Shekhar,95

#Implemneting Bubble sort technique

def marks\_input():

get\_input=input("Enter the names and marks of students separated by comma: ")

data\_list=get\_input.split(',')

Namelist=[x for x in data\_list[0::2]]

Markslist=[x for x in data\_list[1::2]]

# Check len of both list is equal, check name list is all alphabets and marks list all num.

return (Namelist,Markslist)

#Name list: ['RAM', 'Rohit', 'Manali', 'Sonali', 'Shekhar']

#Markslist is: ['80', '90', '100', '85', '90']

def Ranking\_system(\*args):

for i in range(0,len(Marks)-1):

for j in range(0,len(Marks)-1-i):

if Marks[j] < Marks[j+1]:

Marks[j],Marks[j+1]=Marks[j+1],Marks[j]

Names[j],Names[j+1]=Names[j+1],Names[j]

print (Names)

print (Marks)

return (Names,Marks)

Names,Marks=marks\_input()

print ("Your names and marks are: ", Names,Marks)

Finak\_ranks=Ranking\_system(\*Names,\*Marks)

print ("Your final ranks are: ")

for i in range(0,len(Names)):

print ("Rank",i,":",Names[i])

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the names and marks of students separated by comma: RAM,80,Rohit,90,Manali,70,Sonali,85,Shekhar,95

Your names and marks are: ['RAM', 'Rohit', 'Manali', 'Sonali', 'Shekhar'] ['80', '90', '70', '85', '95']

['Shekhar', 'Rohit', 'Sonali', 'RAM', 'Manali']

['95', '90', '85', '80', '70']

Your final ranks are:

Rank 0 : Shekhar

Rank 1 : Rohit

Rank 2 : Sonali

Rank 3 : RAM

Rank 4 : Manali

# Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

import datetime

a=input("Enter your name and age separated by space: ")

lista=a.split()

b=100-int(lista[1])

now = datetime.datetime.now()

year\_of\_100=int(now.year)+b

print ("Hi", lista[0],"your age will be 100 years in", year\_of\_100)

print (dir(datetime.datetime))

# The Program when run should ask for the first name of day a person and on the correct name input, should print out the Birthday of the same person. And if the birthday is not found should print out the appropriate message %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

my\_dict={'Manali':'17may','Sonali':'13oct','Shekhar':'22sep','Shivam':'5oct'}

a=input("Enter teh name who's bday yu wish to see: ")

try:

print ("The Birthday of",a,"is",my\_dict[a])

except keyerror:

print ("This person's bday is not known")

Method2: ---------------------------------------------------------------------------> Use of format function

birthday={'Manali':'17may','Sonali':'13oct','Shekhar':'22sep','Shivam':'5oct'}

name=input("Enter teh name who's bday yu wish to see: ")

if name in birthday.keys():

print("{}'s birthday is on {}".format(name,my\_dict[name])

else:

print("Sorry birthday not known for {}".format(name))

with Format function you don't worry about the cast an int as a string while printing.

# Accept string from the user and display only those characters which are present at an even index

%%%%%%%%%%%%%%%%%%%%%%%

a=input("Enter teh string: ")

b=len(a)

print("The string even chars are: ",a[0:b:2])

# Given two lists of integers; create a third list such that it should contain only odd numbers from the first list and even numbers from the second list %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=[1,2,3,4,5,6]

b=[7,8,9,10,11,12]

j=0

k=0

lista=[]

for i in a:

if (i%2==0):

lista.append(i)

j=j+1

else:

continue

for i in b:

if (i%2==0):

continue

else:

lista.append(i)

j=j+1

print ("The final list is: ",lista)

output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

The final list is: [2, 4, 6, 7, 9, 11]

# How to print the string :"yadoT si a doog yad" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Method1:

a=input("Enter a sentence: ")

list1=a.split()

for i in list1:

print (i[::-1],end=' ')

b=''.join(list1)

Method2:

a=input("Enter a sentence: ")

list1=a.split()

list2=[]

for i in list1:

i=i[::-1]

list2.append(i)

b=' '.join(list2)

print (b)

# The Program when run should ask for the first name of a person and on the correct name input, should print out the Birthday of the same person. And if the birthday is not found ask it and update dictionary %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Method1:

birthday={'Manali':'17may','Sonali':'13oct','Shekhar':'22sep','Shivam':'5oct'}

name=input("Enter the name who's bday yu wish to see: ")

if name in birthday.keys():

print("{}'s birthday is on {}".format(name,birthday[name]))

else:

b=input("Sorry birthday not known ,Please enter the birthday: ")

birthday[name]=b

print ("Final dictionary is:",birthday )

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the name who's bday yu wish to see: Shimi

Sorry birthday not known ,Please enter the birthday: 5oct

Final dictionary is: {'Manali': '17may', 'Sonali': '13oct', 'Shekhar': '22sep', 'Shivam': '5oct', 'Shimi': '5oct'}

# All the numbers in range 1-100 both included, write a list that is perfect squares. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a=[x\*x for x in range(12) if x\*x <= 100]

print (a)

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

# More list comprehensions.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

print ([x\*x for x in range(1,10)])

[x\*\*2 for x in range(10)]

Output: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

[x\*\*2 for x in range(10) if x%2==0]

Output: [0, 4, 16, 36, 64]

myList = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[(lambda x: x\*x)(x) for x in myList]

Output: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

a=[1,2,3,4,5,6,7,8,9]

list=[x for x in a]

list=[x\*x for x in a]

list=[x for x in a if x%2==0]

list=[x\*x for x in a if x%2==0]

list=[x\*x for x in a if x%2!=0]

a=[x\*x for x in range(12) if x\*x <= 100]

# Write a function which accepts the default arguments %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

def greeting(name,greet = 'day'): -----------------------------------> Use of format function and f string

statement=f' Hi {name}!, Good {greet} '

# statement=' Hi {}!, Good {} '.format(name,greet)

print (statement)

greeting('Manali','morning')

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Hi Manali!, Good morning

# Ask for a string from the user and tell him if it is a palindrome or not. Your program should work on numeric data as well

Examples of a Palindrome: "level", "madam", 1234321 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

def palindrome(a):

if a[::-1]==a:

return (True)

else:

return (False)

my\_str=input("Enter a string you want to check for palindrome: ")

if palindrome(my\_str)== True:

print("String is palindrome")

else:

print("Sring is not palindrome")

# Ask the user input to formulate a list of integers and print the output as a new list whose elements are the consecutive sums of itself and all the previous elements of the input list: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#Example Input : [1,2,3,4,5] O/P: [1,3,6,10,15]

#a[0]=a[0]

#a[1]=a[1]+a[0]

#a[2]=a[2]+a[1]

#a[n]=a[n]+a[n-1]

def my\_sum(\*nums):

nums\_new=[int(x) for x in nums]

for i in range(len(nums\_new)-1):

nums\_new[i+1]=nums\_new[i+1]+nums\_new[i]

return (nums\_new)

list=input("Enter the list of numbers to be added separated by space: ")

num\_list=list.split()

result\_list=my\_sum(\*num\_list)

print(f'The original list is {num\_list}, final list is {result\_list}')

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the list of numbers to be added separated by space: 1 2 3

The original list is ['1', '2', '3'], final list is [1, 3, 6]

# Ask the user for two strings and let her know that the two inputs are anagrams or not: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Example of an anagram: LISTEN and SILENT

def anagram(a,b):

set1=set([i for i in a])

set2=set([i for i in b])

if set1 == set2:

return (True)

else:

return (False)

str\_input=input("Enter two strings to be checked for anagram separaed by comma: ")

str1,str2=str\_input.split(',')

if anagram(str1,str2)== True:

print("The strings entered are anagram")

else:

print("The strings entered are not anagram")

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter two strings to be checked for anagram separaed by comma: LISTEN,SILENT

The strings entered are anagram

# Ask the user for name and marks of 5 students and arrange them in order of the rankings

i.e. the one with the highest marks gets the Rank 1, the second highest, Rank2 and so on. In case there is a tie between two students the Name in alphabetical order is the tiebreaker. For examlple, if Ram and Rohit have 80 marks each then Ram gets a rank higher than that of Rohit's %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

---------------------------------------> Implement this for same number of marks and

#RAM,80,Rohit,90,Manali,70,Sonali,85,Shekhar,95

def marks\_input():

get\_input=input("Enter the names and marks of students separated by comma: ")

data\_list=get\_input.split(',')

Namelist=[x for x in data\_list[0::2]]

Markslist=[x for x in data\_list[1::2]]

return (Namelist,Markslist)

#Name list: ['RAM', 'Rohit', 'Manali', 'Sonali', 'Shekhar']

#Markslist is: ['80', '90', '100', '85', '90']

def Ranking\_system(\*args):

for i in range(0,len(Marks)-1):

for j in range(0,len(Marks)-1-i):

if Marks[j] < Marks[j+1]:

Marks[j],Marks[j+1]=Marks[j+1],Marks[j]

Names[j],Names[j+1]=Names[j+1],Names[j]

print (Names)

print (Marks)

return (Names,Marks)

Names,Marks=marks\_input()

print ("Your names and marks are: ", Names,Marks)

Finak\_ranks=Ranking\_system(\*Names,\*Marks)

print ("Your final ranks are: ")

for i in range(0,len(Names)):

print ("Rank",i,":",Names[i])

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Enter the names and marks of students separated by comma: RAM,80,Rohit,90,Manali,70,Sonali,85,Shekhar,95

Your names and marks are: ['RAM', 'Rohit', 'Manali', 'Sonali', 'Shekhar'] ['80', '90', '70', '85', '95']

['Shekhar', 'Rohit', 'Sonali', 'RAM', 'Manali']

['95', '90', '85', '80', '70']

Your final ranks are:

Rank 0 : Shekhar

Rank 1 : Rohit

Rank 2 : Sonali

Rank 3 : RAM

Rank 4 : Manali

# Given a list of integers find the maximum product taking three elements at a time For Example: [12,45,22,1,6,7]--> O/p: 11880(12\*45\*22)

Method1:

list1=[12,45,22,1,6,7]

list2=[]

for i in range(0,len(list1)-2):

list2.append(list1[i]\*list1[i+1]\*list1[i+2])

print (list2)

print (list2)

print (max(list2))

Output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

[11880]

[11880, 990]

[11880, 990, 132]

[11880, 990, 132, 42]

[11880, 990, 132, 42]

11880

Method2:

def Product\_adjacent(\*args):

list1=list(args)

list2=[]

for i in range(0,len(list1)-2):

list2.append(list1[i]\*list1[i+1]\*list1[i+2])

return (max(list2))

list1=[12,45,22,1,6,7]

print ("Maximum product is : " , Product\_adjacent(\*list1))

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Maximum product is : 11880

# Use the Input String: "Led 34215 Zeppelin 5620 is 384 the 143 greatest 789 band 876 ever"-->

O/p Led Zeppelin is the greatest band ever | 011223334445566778889. To make it more interesting do it in 2 lines of code

where is this coming from "011223334445566778889"---???

list1="Led 34215 Zeppelin 5620 is 384 the 143 greatest 789 band 876 ever".split()

list\_alpha=list(filter(lambda x:isdigit(x) == False,list1))

list\_digits=list(filter(lambda x:isdigit(x) == True,list1))

print (' '.join(list\_alpha), "|", ''.join(list\_digits))

output:

C:\Users\GS-0838\Desktop\Roadmap\Python>python .\temp.py

Led Zeppelin is the greatest band ever | 342155620384143789876

Homework by Mayank :-

Problem 1: Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old

Problem 2: The Program when run should ask for the first name of a person and on the correct name input, should print out the Birthday of the same person. And if the birthday is not found should print out the appropriate message

Problem 3: Accept string from the user and display only those characters which are present at an even index

Problem 4: Given two lists of integers; create a third list such that it should contain only odd numbers from the first list and even numbers from the second list  
  
Problem 5: Think of a Problem and code a solution for it.

And Don't forget:**"yadoT si a doog yad"**

1) Ask for a string from the user and tell him if it is a palindrome or not. Your program should work on numeric data as well

Examples of a Palindrome: "level", "madam", 1234321

2) Ask the user input to formulate a list of integers and print the output as a new list whose elements are the consecutive sums of itself and all the previous elements of the input list:

Example Input : [1,2,3,4,5] O/P: [1,3,6,10,15]

3) Ask the user for two strings and let her know that the two inputs are anagrams or not: Example of an anagram: LISTEN and SILENT

4) Ask the user for name and marks of 5 students and arrange them in order of the rankings i.e. the one with the highest marks gets the Rank 1, the second highest, Rank2 and so on. In case there is a tie between two students the Name in alphabetical order is the tiebreaker. For examlple, if Ram and Rohit have 80 marks each then Ram gets a rank higher than that of Rohit's

Here is a quick link to have pip installed on your system:

<https://www.liquidweb.com/kb/install-pip-windows/>

And here is the good stuff:

Problem 1: Given a list of integers find the maximum product taking three elements at a time For Example: [12,45,22,1,6,7]--> O/p: 11880(12\*45\*22)

Problem 2: Use the Input String: "Led 34215 Zeppelin 5620 is 384 the 143 greatest 789 band 876 ever"-->

O/p Led Zeppelin is the greatest band ever | 011223334445566778889. To make it more interesting do it in 2 lines of code

Problem 3: Given a list of integer find the pairs of elements that adds up to a particular value: Example: [1,2,3,4,5,6,7,8,9] Sum=10 O/P: (1,9),(2,8),(3,7),(4,6)

Note both the list and the sum value has to be taken as an input from the user

Good Luck, and don't forget to mail over your solutions to me before we meet tomorrow.

Do ask if any of the problem statement is not clear.

Also, don't forget to try a try block with except and finally at the same time.

And, yes we can't break out of an if block, that was a miss on my part. Well, you learn something new every day. Thanks for pointing it out.☺

Please forward this to anyone I might have missed. Till Next time.

Homework 28-06-2019

input list 1 to 300

write a list comprehension in that

list output gives a number which are multiple of 15

efore we meet tomorrow, I want you guys to have some more practice.

1) List comprehension to get all the numbers in the range 1,300 which have the sum of digits as 10  
2) In a given input string find the longest palindrome--> Sample input: "aabcabaab" O/P: 'baab'  
3) Find all the prime numbers in range 1 to 500  
4) Reverse a number without converting it into a String--> Sample input--> 4567 O/P: 7654  
5) Print the first non-repeated character in a string. Sample Input: "India will win the cricket world cup 2019" O/P: 'a'  
6) From a list of consecutive integers find the missing number-->Sample input: [31,32,34,35,37] O/P--> 36