1. Write a Python program to find the sum of the series 1!/1 + 2!/11 + 3!/111 + 4!/1111 + … up to
n terms. Take n as input.

Sample Input & Output

Input	Output
4	1.2124455688812124
6	1.2129495734352576

2. Write a Python program that will take a number and return whether the number is a Kaprekar number or not.

A Kaprekar number is a number whose square can be divided into two parts such that the sum of those two parts is equal to the original number and none of the parts has a value of 0.

Sample Input & Output

Input	Output
9	9 is a Kaprekar number
45	45 is a Kaprekar number
297	297 is a Kaprekar number

Explanation: $9^2 = 81$

8 + 1 = 9

So, 9 is a Kaprekar number.

Again, $45^2 = 2025$

20+25 = 45

So, 45 is a Kaprekar number.

Similarly, $297^2 = 88209$

88+209 = 297

297 is a Kaprekar number.

3. Given a circle and a line segment. Find out which one of the following is true.

- The line segment intersects the circle.
- The line segment is outside the circle.
- The line segment touches the circle.

Sample Input & Output

Input	Output
0 0 10 0 0 2 2	The line segment intersects the circle
4 3 10 100 100 200 200	The line segment is outside the circle.
0 0 20 -20 -100 -20 100	The line segment touches the circle.

Instruction: The program will give two input prompts to the user. Firstly, it will ask for the circle's center coordinates and the radius. Secondly, another input prompt will be shown to the user to input the coordinates of the two endpoints of the line segment.

- If the distance of the line segment's middle point from the circle's center is less than the radius then the line intersects the circle.
- If the distance of the line segment's middle point from the circle's center is greater than the radius then the line is outside the circle.
- If the distance of the line segment's middle point from the circle's center is equal to the radius then the line touches the circle.

Explanation: In the first sample 0 0 is the coordinate (x,y) of the circle's center and 10 is the radius. In the second line, 0 0 is the start point (a,b) of the segment and 2 2 (c,d) is the endpoint of the segment.

4. Write a Python program that takes a positive integer as input and removes all the digits that are greater than 9 minus the digit to their left.

Sample Input & Output

Input	Output
2572908	25205
4567	45
1234	1234

5. Write a Python function that will take an integer as an argument. If the sum of the squared digit of the number eventually becomes 1 then the function will return 1. Otherwise it will return 0. See the samples for clarification.

Sample Input & Output

Function call	Output
squared_one(203)	1
squared_one(1)	1
squared_one(101)	0

Explanation: $203 : 2^2 + 0^2 + 3^2 = 13$

$1^2 + 3^2 = 10$

$1^2 + 0^2 = 1$

Hence, The function should return 1.

On the other hand, $101 : 1^2 + 0^2 + 1^2 = 2$

So, the function should return 0 for this number as an argument.

6. Write a Python program that takes a positive integer N as input and adds 1 to each of its even digits and removes each of its odd digits.

Sample Input & Output

Input	Output
834	95
753	0
745902	513

7. Look at the series below:

1 + ½ + ⅔ + ¾ + ⅘ + ...

Write a Python program that will take an integer (n) as the limit of the series and print the value of the series. The program should also print a dictionary where the value is the numbers (series' elements) and alphabets in proper order as the keys. Since the keys are English alphabets so there's a constraint. The value of n cannot be larger than 26 or less than 1 (1 < n < 27)

Sample Input & Output

Input	Output
4	The value of the series is 3.716666666666667 Dictionary: {'a': '1/2', 'b': '2/3', 'c': '3/4', 'd': '4/5'}
7	The value of the series is 6.2821428571428575 Dictionary: {'a': '1/2', 'b': '2/3', 'c': '3/4', 'd': '4/5', 'e': '5/6', 'f': '6/7', 'g': '7/8'}
27	The valid range is 1 < n < 27 Please try again
0	The valid range is 1 < n < 27 Please try again

8. A series is given.

1 + 3/2 + 9/3 + 27 + 81/2 + 243/3 + 729 + ...

Write a Python program to get the nth term of the series. N will be inputted by the user.

Sample Input & Output

Input	Output
2	3/2
5	81/2
7	729

9. Write a Python program that takes an integer as input and outputs the adjacent of 5 from the inputted integer.

Sample Input & Output

Input	Output
3456	6
1234	Not found
55879856	5

10. Write a Python program that takes an integer of more than 2 digits and powers the digit in the following manner.

Sample Input & Output

Input	Output
1234	14916
767	493649

Explanation: $1234\text{: } 1^2 = 1,\ 2^2 = 4,\ 3^2 = 9,\ 4^2 = 16.\ \textit{Final output } 14916$

11. Create a Python function named **dcd()** to decode some specific encrypted message. Sample messages and their meaning is given. Make a function so that a user passes the secret messages to the function's parameter and it prints the decoded message.

Sample Call & Output

Function call	Output
dcd('Wt|fs%nx%ymj%nrutxyjw')	Rowan is the imposter
dcd('Jsjr~%fy%stwym3%^tz,wj%hqjfwji%mty3')	Enemy at north. You're cleared hot.

12. Create a Python function named **dcn()** to deconcatenate strings. The function should return the deconcatenated string. If the user inserts integers as arguments then the integers will be converted into strings.

Sample Call & Output

Function call	Output
dcn(1234,12)	'34'
dcn('7465','76')	'45'
dcn(qwert,ew)	'qrt'
dcn(qsd,h)	'None'

13. Write a Python function named **evn()** that will take an integer as an argument and return True if it's an even number. Else, it should return False.

Sample Call & Output

Function call	Output
evn(2)	True

evn(3)	False

14. A list of integers and a target is given. If two integers in the list sums up and equals the target, then print their indices.
Sample Given & Output

Given	Output

Nums = [3,2,4]
Target = 6
	[1,2]

Nums = [2,7,11,15]
Target = 9
	[0,1]

Explanation: 2+4 = 6. 2 and 4 respectively staying at 1st and 2nd index (Sample Given 1)

15. Write a Python function named **med()** that takes a list as an argument and returns the median from a list of integers.
Sample Call & Output

Function call	Output
med([1,2,3,4,5])	3
med([4,6,8,0])	7

16. Write a Python function named **str_avg_dct()** that will take a dictionary as an argument. The keys of the dictionary are tuples containing integers and the values are alphabetical strings. The function should return a new dictionary where the values will be the average of the integers (integer converted) inside the tuple and the keys will be the last character of the string.
Sample Call & Output

Function call	Output

str_avg_dct({(2,5,4): ‘asdf’, (1,0,1): ‘Taskforce’, (9,6,0,5): ‘Eight’})	New dictionary: {‘f’: 3, ‘e’: 0, ‘t’: 5}

17. Write a Python program that takes an integer of more than 2 digits and powers the digit in the following manner.
Sample Input & Output

Input	Output
1234	1427256
767	736343

Explanation: $1234\text{:}\ 1^1 = 1,\ 2^2 = 4,\ 3^3 = 27,\ 4^4 = 256.\ \textit{Final output}\ 1427256$

18. Write a Python function that will take a numeric value as an argument and return an alphabet. The numeric values start from 26 and end at 1 unlike the ASCII table. From 26 to 1 the alphabets are a to z. Name the function **num_to_alph()**
Sample Call & Output

Function Call	Output
num_to_alph(26)	‘a’
num_to_alph(25)	‘b’

19. Create a Python function named **sequence_sumsub()** that takes a tuple (or you can use **list** instead, it's up to you. I used tuples.) as an argument and sums and subtracts items sequentially shown in the following manner.
Sample Call & Output

Function call	Output

sequence_sumsub((1,4,3,2))	4
sequence_sumsub((4,4,4,6,2))	8

Explanation: 1+4-3+2 = 4
4+4-4+6-2 = 8

20. Write a Python program where the program will take two inputs from the user. One is an integer and the other is a list of alphabets. The integer is the range of English alphabets. If any alphabet that exists between the range is missing in the inputted list then the program will print the alphabets as output. See samples for clarification. Your program should work for every sample.

Sample Input & Output

Input	Output
a,b,c,d 6	Existing alphabets: [‘a’,’b’,’c’,’d’] Alphabets to add: [‘e’,’f’]
[b,d] 8	Existing alphabets: [‘b’,’d’] Alphabets to add: [‘a’,’c’,’e’,’f’,’g’,’h’]
[g,h] 4	No alphabet is existing between the range Alphabets to add: [‘a’,’b’,’c’,’d’] Alphabets to remove: [‘g’,’h’]

21. A list is given. Create a new list with the items that have ‘a’ on it.

Sample Given & Output

Input	Output
list1 = [‘Apple’, ‘Banana’, ‘Text’, ‘Novice’, ‘Zodiac’]	[‘Apple’, ‘Banana’, ‘Zodiac’]
list2 = [‘1123’, ‘Sun’, ‘Moon’]	[ ]

22. Build a Python function that will take a list of items and concatenate all of the items and return the string. Name the function **str_con().**

Sample Call & Output

Function call	Output
str_con([1,2,3,True,False,0.5])	‘123TrueFalse0.5’
str_con([‘8’,9,int(6.5), 3>2])	‘896True’

23. Python can sum (concatenate) strings and list but they cannot subtract those. Create a Python function with two parameters that will take two list arguments and if any common element exists between the lists, it will make a new list removing the common elements and return the new list. You can name the function **uncommon().**

Sample Call & Output

Function call	Output
print(uncommon([1,2,3,4],[2,3]))	[1,4]
print(uncommon([‘q’, ‘s’, ‘d] ,[‘q’,’s’,’d’])	[ ]
uncommon([5,3,4,8,9],[a,b,3,4])	[5,8,9,a,b]

24. Write a Python function named **sum_sequence()** that will take a list as an argument and return a new list with the summation of the numbers. (See samples). Make sure that your program works for every list. If any item of the list (passed argument) isn’t numeric then the function should return None (no output will be displayed).

Sample Given & Output

Function call	Output
[5,3,4,2,8,7]	[8,12,14,22,29]

[5,8,’a’]	

25. Write a Python program to find out the most appearing substring from several string portions. The strings are separated by a “-” and each portion contains a substring of 8 characters. Each portion starts with a special character and appears several times in a row. You’ve to find out which portion has the most appearance of the special character.

<u>Sample Input & Output</u>

Input	Output
####1234-##567890-######00	Portion 3
$$123456-$$$$$678-$$$fg567	Portion 2

26. You’re given a n x n matrix. Rotate the matrix as shown in the samples. Make a user defined function to solve this problem. You can name this function **rotate_matrix()**

<u>Sample Call & Output</u>

Function call	Output
rotate_matrix([[1,2,3], [4,5,6], [7,8,9]])	[[7,4,1],[8,5,2],[9,6,3]]
rotate_matrix([[4,5,6,7],[3,3,4,5],[4,5,5,5],[2,3,4,1])	[[2,4,3,4],[3,5,3,5],[4,5,4,6],[1,5,5,7]]

27. Write a Python program where you’re given a list of words. The user will input a word with some mistakes and the program will print the word that has the most alphabetical matches with the given word.

<u>Sample Given & Output</u>

Given & Input	Output

given = [‘Evening’, ‘Morning’, ‘Afternoon’, ‘Night’] Input: avaning	Evening
given = [‘Evening’, ‘Morning’, ‘Afternoon’, ‘Night’] Input: Orni	Morning
given = [‘Evening’, ‘Morning’, ‘Afternoon’, ‘Night’] Input: igH	Night

28. Write a Python function named **overloaded()** that will have three parameters. It will take two string arguments and return the multiplied/concatenated number/string. The remaining parameter is a default parameter whose argument is 1.

- If both strings are numeric (integer only) then the program should return the multiple of the numbers. If the default argument is changed then the resultant will be concatenated to the resultant default argument times.
- If both strings are non-numeric then the program should return the concatenated string default argument times.
- If one string is numeric and the other one is non-numeric then the program should return the concatenated string as well. Default argument’s functionality stays the same as mentioned before.

Sample Call & Output

Function call	Output
overloaded(22,3)	‘66’
overloaded(‘abc’,’def’)	‘abcdef’
overloaded(‘a’,’bc’,3)	‘abcabcabc’
overloaded(‘ar’,23)	‘ar23’
overloaded(‘22’,’3’,2)	‘6666’

29. You'll be given a list of pin numbers of 4 digits. Write a Python program that will take a 2 digit integer number and match it with the pin numbers. If every digit of the integer belongs to a certain pin of the list then the program will create a new list for the pin numbers. If the integer contains the same digits then the program will create the new list judging the first digit. Finally, print the list.

given = [2032, 4354, 6821, 4567, 9086, 1023, 1011]

Sample Input & Output

Input	Output
23	[2032,1023]
11	[6281,1023,1011]
97	[ ]

30. Write a Python program that lets a user input the products he wants and the input taking process only stops when he presses Enter without inputting the name of the product. Then count the total price for the products.

Available products: XXX = 250 Taka
YYY = 150 Taka
ZZZ = 70 Taka

Sample Input & Output

Input	Output
XXX XXX YYY	650 Taka
YYY XXX ZZZ	470 Taka

31. Write a Python program that takes a number as input and prints the factorial of that number in the console. Also print the sum of the numbers from 1 to the user input and print their average as well.

Sample Input & Output

Input	Output
5	Factorial: 120 Sum: 15 Average 3.0

32. Write a Python program that will read some words (the first alphabet of each word will be in uppercase) and other alphabets will be in lowercase without any space and make a list with the words.

Sample Input & Output

Input	Output
StealthEthereaJamesDane	['Stealth', 'Etherea', 'James', 'Dane'
RiverCloudSky	['River', 'Cloud', 'Sky']

33. Consider the following series:

1 + 3 + 7 + 15 + 31 + ...

Your program should take an integer n and print the nth number of the pattern. Also print the summation of the series.

Sample Input & Output

Input	Output
3	nth number: 7 Sum: 11
6	nth number: 63 Sum: 120

34. Build a Python function named **str_merger()** that will take two string arguments and return a new string. The new string will consist of the even indexed characters of the first argument and the odd indexed characters of the second argument. In the new string, even indexed characters will be in uppercase and odd indexed characters will be in lowercase.

Sample Call & Output

Function call	Output
str_merger(‘abcdef’,’ertyu’)	‘AcErY’
str_merger(‘Stealth’,’Etherea)	‘SeLhTeE’

35. Strings are immutable in Python. Create a function so that we can make the strings mutable. The function will receive three arguments in the parameters. The first argument will be the string and the second argument will be the index position. The last argument is the character string that we want to replace with the existing character on the index position. You can name this function **str_mutator()**

Sample Call & Output

Function call	Output
str_mutator(‘Etherea’, 6, ‘al’)	‘Ethereal’
str_mutator(‘Heaven’, 5, ‘nly’)	‘Heavenly’
str_mutator(‘Red Dragon’, 3, ‘x’)	‘RedxDragon’
str_mutator(‘River’, 1, 1)	‘R1ver’

36. You’re given a list. Number of the list items are multiples of a random integer n. Write the Python script to make a new list where each sub-list will contain four items consecutively.
Constraint: 1 < n < 5
Sample Given & Output

Given	Output
[1,2,3,4,5,6,7,8,9,10,11,12]	[[1,2,3,4],[5,6,7,8],[9,10,11,12]]
[‘a’,’b’,’c’,’yoink’,’e’,’f’,’g’,’bonk’]	[[‘a’,’b’,’c’,’yoink’],[‘e’,’f’,’g’,’bonk’]]
[1,2,3,5,6,7,8,9,10]	[[1,2,3],[5,6,7],[8,9,10]]

Description: The item number of the first two samples are the multiple of 4. So each sublist carried 4 items. The third sample shows us that the item numbers are multiple of 3. Hence each sublist contains three items. Note that the first two samples can be solved by assuming the item number to be the multiple of 2. For doing this kind of assumption and while working for larger lists, don’t violate the constraint.

37. Write a Python function that will take a dictionary as an argument. Keys of the dictionary are strings and the values are tuples and lists of numbers (integer and floats). Your program should return a new dictionary where there will be only a key and a value. The key is a string holding the ASCII summation of the given dictionary’s key strings and the value is the average of the given dictionary’s values. Name the function **single_dict()**.
Sample Call & Output

Function call	Output
single_dict({‘String’:(2,5,4), ‘Integer’:[10,10], ‘Float’: (6,9,8,13)}	{1851: 7.444}

You don’t need to round the output as shown in the sample output.

38. Tuples are immutable in Python. Create a function so that we can make the tuples mutable. The function will receive three arguments in the parameters. The first argument will be the tuple and the second argument will be the index position. The last argument is the item that we want to replace with the existing character on the index position. You can name this function **tuple_mutator()**

Sample Call & Output

Function call	Output
tuple_mutator((2,3,5,8),2, ‘a’)	(2,3,’a’,8)
tuple_mutator((‘Peace’, ‘Moon’, ‘Celestial’),0, ‘Echo’)	(‘Echo’, ‘Moon’, ‘Celestial’)

39. Python cannot return the keys+value length of a dictionary. Create a user defined function called **hybrid_len()** that can return how many items are there in a dictionary. Make sure that your program works for sublists/subtuples too. Assume that, there can only be a sublist/subtuple inside the value.

Sample Call & Output

Function call	Output
hybrid_len({‘a’: (1,2,[x,y,z],4), ‘b’: [9,8,(y,u,i,o),65,4]}	16
hybrid_len({‘apple’: 4, ‘banana’: 5, ‘mango’: 6}	6

40. Create a function named **dict_to_list()** that will receive a dictionary argument and returns a list with the keys and values of the dictionary. In the list, keys and values should be placed adjacently. Hint: [a,1,b,33,c,55] # pore remove kore dibo

Sample Call & Output

Function call	Output

dict_to_list({‘Pen’: 2, ‘Paper’: 3, ‘Notepad’: 1})	[Pen, 2, Paper, 3, Notepad, 1]
dict_to_list({‘Stx’: (10,20), ‘Eth’: (11,), ‘D9X’: (13,12,16)}	[‘Stx’, (10,20), ‘Eth’, (11,), ‘D9X’, (13,12,16)]

41. Write a Python program that will take a list from the user and print the subtracted value of the numeric items. Note that the list is given. The user need not to input the list to the program.

Sample Input & Output

Given	Output
[‘Lily’, ‘23’, 23, ‘Rose’, ‘1.85’]	-1.85
[‘a’, True, 6<7, ‘5’, 7, ‘8’, 9, 2.5]	-21.5

42. Create a Python function named **list_to_dict** that will take a list made of integer and string data types. The function will return a dictionary where the integer will be the values and the leftover strings will be the keys. If the number of numeric and string items do not match then it would be impossible to make the dictionary. So, put a message regarding that if the user gives invalid input.

Sample Call & Output

Function call	Output
list_to_dict([100, 200, ‘a’, 250, ‘b’, ‘c’])	{‘a’: 100, ‘b’: 200, ‘c’: 250}
list_to_dict([‘100’, 100, 200, ‘d’, ‘x’, 800, 700])	Invalid list. The numbers of strings and integers do not match.
list_to_dict([‘100’, 100, 200, ‘d’, ‘x’, 800])	{‘100’: 100, ‘d’: 200, ‘x’: 800}

43. In the previous problem, you made a dictionary. Now create another function that will do the same thing but the values will be in reverse order. You can name this function **reverse_list_to_dict()**

Sample Call & Output

Function call	Output
list_to_dict([100, 200, ‘a’, 250, ‘b’, ‘c’])	{‘a’: 250, ‘b’: 200, ‘c’: 100}
list_to_dict([‘100’, 100, 200, ‘d’, ‘x’, 800, 700])	Invalid list. The numbers of strings and integers do not match.
list_to_dict([‘100’, 100, 200, ‘d’, ‘x’, 800])	{‘100’: 800, ‘d’: 200, ‘x’: 100}

44. Write a Python function named **reversed_dict()**. A dictionary will be passed to the parameter as the argument and the function will return a new dictionary where the keys will be turned as values and the values will be the keys.

Sample Call & Output

Function call	Output
reversed_dict({‘a’: 35, ‘b’ : 36})	{35: ‘a’, 36: ‘b’}

45. A dictionary is given. Write a Python program that will print a new dictionary where the keys will be the string concatenation of every key and the value will be the string concatenation of the values. If any numeric string (ASCII range 48-57) exists in the given dictionary’s key/value then the numbers will be summed up instead of string concatenation. The summed up numbers will be later concatenated with the string type items. Make sure your program works for every given dictionary.

Constraint: No float numbers can be given as key or value to the given dictionary.

Sample Given & Output

Given	Output
{‘a’: 59, ‘b’: 69, ‘c’: ‘Yes’, ‘d’: True, ‘e’: ‘49’, 1: ‘No’, ‘2’: False}	{abcd3 : YesTrueNoFalse177}
{‘R’: 1, 2: ‘3’, ‘F’: a}	{RF2 : a1}

46. Python has no built-in function to get a character’s position on a negative indexing system. Write a Python function named neg_idx() that will return the negative index position of a string. There will be two arguments. The first one is for the whole string and the second one is for the character. If any integer/float or other data type is passed to the parameter then they will be converted into strings first.

<u>Sample Call & Output</u>

Function call	Output
neg_idx(‘Mr. String’, ‘S’)	-6
neg_idx(‘A rabbit’, ‘ ‘)	-7
neg_idx(1234, 3)	-2

47. You’re given two lists. Make two more new lists with squaring the numbers from the given lists. Then, sum the even indexed numbers of the first new list and subtract the odd indexed numbers of the second new list. You’ll get two resultants from the new lists. Then sum the two resultants. Finally, print the sum.

<u>Sample Given & Output</u>

Given	Output

List1 = [2,4,6,8]
List2 = [1,9,5,7]
	72

List1 = [10,11,14,15,20]
List2 = [21,15,18,22]
	437

List1 = [1,1,1,1]
List2 = [1,1,1,2]
	-1

48. You’re given a dictionary. Print the key with the biggest length.

Sample Given & Output

Given	Output
{‘Serena’: 18, ‘Etherea’: 19, ‘Stealth’: 21, ‘Dane’: 25, ‘Ghost’: ‘35+’}	Etherea
{‘DS E-Tense’: 1, ‘Chevrolet’: 2, ‘Saleen S1’: 3, ‘Porsche’: 5, ‘BMW’: 6}	DS E-Tense

49. Write a Python program that will take a string (single length) and print the previous and next five characters of the string from the ASCII table.

Sample Input & Output

Input	Output
‘f’	Previous 5: [‘e’, ‘d’, ‘c’, ‘b’, ‘a’] Next 5: [‘g’, ‘h’, ‘i’, ‘j’, ‘k’]

50. A dictionary will be given to you which consists of single lengthed strings as keys and values are tuples with integers. But be aware that the tuples might have some strings inside them too. Write a Python program that will create a new dictionary where the keys will stay the same (There’s a miscellaneous case by the way) . In the new dictionary, the values of the corresponding keys will be either the average or the summation of integers inside each tuple. If the tuple length is an even integer then the value of the key will be the summation of the integers. If the tuple length is an odd integer then the value of the key will be the average of the integers. Moreover, if any string exists in any of the tuples, the string will be concatenated with the key. Your program should work for every given list.

Sample Given & Output

Given	Output
{‘a’: (5,10,5), ‘b’: (11,12,13,14), ‘c’: (9,8,3,’a’), ‘d’: (3,4,3,9,5,’x’,’y’)}	{‘a’: 6.66, ‘b’: 50, ‘c-a’: 20, ‘d-yd-x’: 4.8}
{‘a’: (5,10,5), ‘b’: (11,12,13,14), ‘c’: (9,8,3,’a’), ‘d’: (3,4,3,9,5,’x’,’y’,’z’)}	{‘a’: 6.66, ‘b’: 50, ‘c-a’: 20, ‘d-zd-yd-x’: 24}
{‘a’: (5,10,5), ‘b’: (11,12,13,14,’0’,’8’,’a’), ‘c’: (9,8,3,’a’), ‘d’: (3,4,3,9,5,’x’,’y’,’z’)}	{‘a’: 6.66, ‘b-ab-8b-0’: 12.5, ‘c-a’: 20, ‘d-zd-yd-x’: 24}

Note: You may ignore the output sample for averages. You don’t need to round the average value. Keep it as it is.

51. Write a Python program that can do the following task shown in the explanation.

Given	Output

list1 = [10,20,15,40]
list2 = [12,22,42,32]
	193

list1 = [11,12,21,14,15,16]
list2 = [2,6,5,3,9,10]
	124

Explanation:

Given 01:

list1 operation

10 + 40 = 50

20 + 15 = 35

list1_new = [50,35]

list2 operation

12+32 = 44

22+42 = 64

list2_new = [44,64]

final_sum = 50+35+44+64 = 193

Given 02:

list1 operation

11 + 16 = 27

12+15 = 27

21 + 14 = 35

list1_new = [27,27, 35]

list2 operation

2+10 = 12

6+9 = 15

5+3 = 8

list2_new = [12,15,8]

final_sum = 27+27+35+12+15+8 = 124

Constraint: The given lists' length cannot be an odd integer and the items can be only integers.

52. Create a Python function named **str_checker**. The function will receive a list/tuple as an argument and return True if the sequence has any string data type. Else, it will return False.

Sample Call & Output

Function call	Output
str_checker((1,2,3,'t', 3>2))	True
str_checker([True, 2.5, 100])	False

53. Create a Python function named **middle()** that will return the middle item of a list or tuple. If the length of the list is an even integer, then the middle item can't be found. So the function will return the concatenated strings of pre-middle and post-middle items.

Sample Call & Output

Function call	Output

middle([1,2,3,4])	‘2-3’
middle((‘Abs’, ‘len’, ‘append’, ‘max’, ‘min’))	‘append’
middle((‘Newton’, ‘Einstein’, ‘Rutherford’, ‘Bohr’))	‘Einstein-Rutherford’

54. Write a Python program that will take a word from the user and print the ASCII summation of the lowercase characters and the ASCII multiplication of uppercase characters. These data will be printed via a dictionary. Keys will be the first and last alphabets of the input respectively. See samples for details.

Sample Input & Output

Input	Output
DraGOn	{‘D’: 321, ‘n’: 381412}
StArs	{‘S’: 345, ‘s’: 5395}
lost	{‘l’: 450, ‘t’: 0}
PEN	{‘P’: 0, ‘N’: 430560}

55. Write a Python program that will read dash separated integers from a user and print the characters of the integers from the ASCII table.

Sample Input & Output

Input	Output
97-98-99-100	abcd
65-109-97-120	Amax
41-33-38-35	)!&#

56. Write a Python program which reads a string from the user and prints a new string reversing the even indexed characters.

Sample Input & Output

Input	Output
Notepad	dopetaN
Whale	ehalW

Explanation: (Sample 1) The even indexed alphabets are N,t,p and d. These alphabets replaced their places with each other reversely. As a result, d and N replaced their places, p and t replaced their places.

57. You'll be given a dictionary. Write the Python program to print a single sub-dictionary of the given dictionary that contains the longest (largest length) value. You can use the built-in **max()** function for this task if you want.
Sample Given & Output

Given	Output
{'a': ('*', '-'), 'x': ['a', 'q', 'u', 'A'], 'Num': (2,3,4)}	{'x': ['a', 'q', 'u', 'A']}
{1: [10,20,30], 3: [20,30,40,50], 2: (1,)	{3: [20,30,40,50]}

58. Dictionary items are unordered. Which means, they aren't accessible by indexing. Dictionaries have no attribute called indexing. Since, we can't do indexing on dictionary items, create a user defined function named **dict_idx()** that will take a dictionary and the index number in a parameter. After that, the function should return a single lengthened dictionary (only one key,value pair) that was existing on the index position.
Rules: While making the function, don't forget to count the index from 0. The index will increment by +1 after passing each key,value pair. (Just like the indexes of sequence data types)
Sample Call & Output

Function call	Output

dict_idx({‘a’: (‘*’, ‘-’), ‘x’: [‘a’, ‘q’, ‘u’, ‘A’], ‘Num’: (2,3,4)},2)	{‘Num’: (2,3,4)}
dict_idx({1: [10,20,30], 3: [20,30,40,50], 2: (1,)},0)	{1: [10,20,30]}

59. A list will be given to you which contains some string elements. Some of the strings are concatenated with each other. They are concatenated by a single dash. Write a Python program that will decompose the concatenated strings and print a new list with them.

Sample Input & Output

Input	Output
[‘CSE110’, ‘CSE111-CSE230’, ‘ENG101’, ‘ENG102-PHY111’]	[‘CSE110’, ‘CSE111’, ‘CSE230’, ‘ENG101’, ‘ENG102’, ‘PHY111’]
[‘Cat-Dog’, ‘Apple-Banana’, ‘Pen’, ‘Paper’, ‘PDF’]	[‘Cat’, ‘Dog’, ‘Apple’, ‘Banana’, ‘Pen’, ‘Paper’, ‘PDF’]

60. Create a user defined function named **str_to_int()**. A list will be passed as the argument that contains the mixture of string, integers, float numbers and boolean data type. The function will return a new list converting every existing element into integers. If the given list item isn’t integer already, then the program should convert it into a string first. After that, it will get the string’s ASCII summation and place it in the new list.

Sample Call & Output

Function call	Output
str_to_int([100, 20, ‘abc’, ‘30’, True, 2>3,])	[100, 20, 294, 99, 416, 491]
str_to_int([‘***’, 19.5, 5==7, 20, True + 1, False+3])	[126, 205, 491, 20, 2, 3]

Note: Expressions are solved first if they exist inside the data structures. 2>3, True + 1, False + 1 won’t show similar behavior like integers or strings. Rather, they’ll be treated like False (boolean) , 2 (integer) and 3(integer) respectively. One more thing, when True and False are placed inside an arithmetic expression, the values for True becomes 1 and False becomes 0.

61. Write a Python program which will take a sentence from the user and print the length of the last word.
Constraint: There can be no punctuations in the sentence and only one whitespace between the words can exist.
Sample Input & Output

Input	Output
Hi my name is Dane	Last word: Dane Length: 4
He is my best friend	Last word: friend Length: 6

62. Write a Python function named **spc_remover()** that will take a sentence as an argument. There will be arbitrary spaces in the sentence. Your program should replace the multiple spaces with a single space.
Sample Call & Output

Function call	Output
spc_remover(‘His name is Bill’)	His name is Bill
spc_remover(‘No mercy’)	No mercy

63. Write a Python program that will keep printing numbers from 1 to n. The program will take input from the user. No matter what the user inputs, for his/her every input your program should print numbers consequently. When the user will press Enter, the program will be stopped and display the dictionary of the user’s entered inputs and the outputs.
Sample Input & Output

Input	Output
A b fgfg etwdgd lll	1 2 3 4 5 {‘A’:1, ‘b’: 2, ‘fgfg’: 3, ‘etwdgd’: 4, ‘lll’:5}

64. Write a Python program that will take an input from the user and display whether it is a number, word or mixed with digit and letters.
If all the characters are numeric values, print DIGIT. If they are all letters,
print ALPH. If it is mixed, print MIXED.
Constraint: Only numbers and alphabets can be inputted.
Sample Input & Output

Input	Output
65Yu	MIXED
0101	DIGIT
Panda	Word

65. Write the Python script of a program which reads a list of words from the user, there are two uppercase alphabets in the words. Your program should print the lowercase letters existing between the two uppercase letters. If there’s no lowercase letters between the uppercase letters then the program will print ‘Not found’.
Constraint: Anything except alphabets is prohibited to input. There must be two uppercase letters in the input.
Sample Input & Output

Input	Output
[DarknesS, riSE, sHootEr, peACe]	(arknes, Not found, oot, Not found)

66. A strong password must contain the following things:

- Uppercase letters (A-Z)
- Lowercase letters (a-z)
- Digits (0-9)
- Special characters (ASCII decimal range 33-47, 64, 94-95, 126)
- Length > 6

Write a Python program that will take a random password from the user and print how strong the password is.

Every criteria satisfies	Password Strength: Very Strong
One criteria misses	Password Strength: Strong
Two criteria misses	Password Strength: Good
More than two criteria misses	Password Strength: Weak
If the length is less than 6	Password Strength : Very Weak (No need to judge other criterias)

Sample Input & Output

Input	Output
aZ#9	Very Weak
aZ#9top	Very Strong
kiroalexa	Weak
kirOaleXa	Good
kirOaleXa#	Strong

67. Write a Python program that gives N times input prompt to the user and then displays a list with the inputted numbers subtraction. N will be given by the user as well. See samples for clarification.

Input	Output

4
6 7 8
10 9 5 7
9 9 9
13 21 32
	
	-9
	-11
	-9
	-40
	[-9, -11, -9, -40]

3
1 2 3 4 5
4 3 2
1 3
	
	-13
	-1
	-2
	[-13, -1, -2]

68. Write a python program that takes a number sequence as input and prints whether it is a UB Jumper or Not UB Jumper. Input receiving process will stop after getting “BREAK” as input.

Concept: Let there are N numbers in a list and that list is said to be a UB Jumper if the absolute values of the difference between the successive elements take on all the values 1 through N – 1. For example, 2 1 4 6 10 is a UB Jumper because the absolute differences between them are 1 3 2 4 which is all numbers from 1 to (5 - 1) or 4.

You cannot use **abs()** function to complete this objective.

Sample Input & Output

Input	Output
1 4 2 3 2 1 4 6 10 1 4 2 -1 6 BREAK	UB Jumper UB Jumper Not UB Jumper

69. You're given a list of words. Some of the words are mixed with uppercase and lowercase letters. Write a Python program that will return a dictionary consisting of the corrected words as keys and list of uppercase and lowercase letters as values.

Sample Given & Output

Given	Output
['apple', 'baNANA', 'pineapple, 'wAtErMelON', MANgo']	{'banana': ['ba', 'NANA'], 'watermelon': ['wtrel', 'AEMON'], 'mango': ['go', MAN']}
['fRUits', 'auDaCITy', 'glasses', 'temPo']	{'fruits': ['fuits', 'RU'], 'audacity': ['auay', 'DCIT', 'tempo': ['temo', 'P']

70. Write a Python program that will read four strings from the user and concatenate the common characters of the four strings. The concatenated strings will be displayed as output. The strings are mixtures of numbers and letters only. Your program should not be a case-sensitive program. If any letters are in uppercase then convert them into lowercases.

Sample Input & Output

Input	Output
Hitman007 HellDane9X StealthAX Eth-01	ha
78Thabc 87hatCB a97O8BdT6 117a8thCCb	78abt

71. Write a Python program that will read (given) a list from the user and print a dictionary. See the sample input and output below.

Sample Given & Output

Given	Output
list1 = [‘Alex’: 400, ‘Dane’: 150, ‘Rin’: 200]	{‘Alex’: 400, ‘Dane’: 150, ‘Rin’:200}
alist = [‘Jones’: 100, ‘Jack’: 25, ‘Riyan’: 45, ‘Lee’: 250]	{‘Jones’: 100, ‘Jack’: 25, ‘Riyan’: 45, ‘Lee’: 250}

72. Write a Python program that will read (input) a list from the user and print a dictionary. See the sample input and output below.

Sample Given & Output

Input	Output
[‘Alex’: 400, ‘Dane’: 150, ‘Rin’: 200]	{‘Alex’: 400, ‘Dane’: 150, ‘Rin’:200}
[‘Jones’: 100, ‘Jack’: 25, ‘Riyan’: 45, ‘Lee’: 250]	{‘Jones’: 100, ‘Jack’: 25, ‘Riyan’: 45, ‘Lee’: 250}