ANAGRAMS 04

Welcome again to the programming screen cast of anagrams in this particular programming screen cast i will be discussing another way to figure out whether two strings are anagrams or not. In the previous programming screen cast we have already discussed a way to figure out whether two strings are anagrams or not, in this particular programming screen cast i will be discussing another way this particular method requires the use of sorted function. So what is sorted function? Sorted basically sorts any sequence list for example list and returns the sorted list so i will show you how can you use the sorted function for example i have a list here named x of integer values and the values present in this is not sorted for example they are four three one two now if i write print sorted of x see it printed the sorted value and it printed the sorted values in ascending order so sorted function basically returns the sorted values in ascending order i repeat sorted function by default sorts the value present in the list in ascending order so id i want to sort the list in descending order what should i do? Well we have a way here, how can you do that? For example we have sorted x and this will pass another parameter named reverse and in this i will write true so let us check whether it works or not yes it works, so if i write reverse is equal to true then it will sort the values in ascending order so i repeat sorted functions sorts the value in ascending order and if i write reverse is equals to true it will sort the values in descending order this is how you can use the sorted function, you can also use the sorted function on different data types for example we can use the sorted function on character values too for example i have x as the list of characters as q for example w e r t and y now if i write print sorted of x press enter so as you see it has sorted the values in alphabetical order so first of all we have e then q r t w and y so if we have list of characters then sort it functions sorts the character in alphabetical order and you can also use the string in this sorted function let us find out how can you use that for example x is a string here and we have python x is a string which has the value python here if i write again print sorted of x and what should it print, yes it has sorted the characters of python in alphabetical order first of all h then n then n o p t and y so now you see you can use a integer list you can use a character list and you can also use the string and use it in sorted function to sort the values present in the particular list or string and i hope that you are already aware of dictionaries we can also use sorted to sort the dictionary so how can you do that, i will take the dictionary here i will again name it as x so i will write for example q, q is the key here and one is the value then i will take w, w is the key here and two is the value and then i have e here for example and its value is three and let’s take one more value i will just write four here and then i will take t here its value is zero so we have a invalid syntax here let us find out where is the problem, so the problem is we didn’t use the curly braces yes in dictionary we should use the curly braces and in list we use the square the rectangle braces so we should use the curly braces here and then if i write print sorted or x see it is sorted the values in alphabetical order again first of all we have e then q then t and w so it is sorting the dictionary values on the basis of keys yes it is sorting the dictionary values on the basis of keys please note this fact now i have another example here for example i have a list of strings example i have ccccc bb then a then ddd and i want to sort this list on the basic of the length of this particular string yes i want to sort this list on the basis of length of particular list so the output should be first of all since a is the list since a is the string in the smallest length so first of all a should be printed then bb should be printed then ddd then ccccc should be printed so how can you do that? This is fairly easy, you just need to write print sorted capital l and the key should be length just write length here and i press enter here i have the desired output i sorted this list of string on the basis of length of the particular string so first of all i have a here then bb then ddd then ccccc so this is how you can sort this on the basis of length of the particular string so i hope now sorted function is clear to you guys now i will use the sorted function in our program in our program to figure out whether two strings are anagrams or not so first of all we should take two strings i will take them str1 and str2 i will just write input enter the first string then i should write str2 is equal to input enter the second string now you can figure out what purpose would sorted function would serve what purpose would sorted function serve here so we can sort this to list str1 and str2 and if the sorted version of these two list is same if the sorted version of these two list is same then you can say that str1 and str2 are anagrams otherwise they are not what should i write here? If sorted of str1 is equal to is equal to sorted of str2 then i should print these are anagrams else i should print these are not anagrams so let us run this program i just save it as anagrams two dot py enter the first string i will enter listen, enter the second string i will enter silent these are anagrams so our output is correct if i run it again enter the first string hello enter the second string thyui any random string these are not anagrams so our output is correct so i will also show you whether the sorted version of these two list are same so i will just print sorted str1 and then print sorted str2 so now let us try to run this again so i will take another anagram example i will take tar and i will take rat sorry i typed it as r a y so these are not anagrams but you see the sorted version are not even same first of all they are a r t then it is a r y now let us check on the list that our anagrams so i will write rat i will write tar so as you can see the sorted version of these two lists are same so first of all let us a r t for rat and for tar it is a r t so for sorted version so these two list is same that's why these are anagram and we can also take the example of two lists of not anagrams so i will just write another list for example how and wow how and wow these are not anagrams since the sorted version of these two list is not same so you can now figure out how you can now figure out how to check whether two strings are anagrams are not and i will just go through the program again first of all you need to input two strings that is fairly simple and then you need to check whether the sorted version of these two list are same, if they are same they are anagrams if they are not same these are not anagrams so this program of anagrams was fairly easy you just need to apply a function and there you are you get the answer so we have discussed two ways whether two strings are anagrams or not there are i think many other ways through which you can figure out i like you to discuss on discussion form and bring out some most nice ideas to check whether two strings are anagrams or not. I hope this programming screen cast was useful to you guys if you have any problem any doubt please refer the discussion form we are there to help. Thank you.