**Python Practical File**

**School of CA & IT**

**(Python Programming BCA-DSE-1.1)**

**Sgrr University**



Submitted by :

Anuj Panthri

BCA V

R190529014

**Index**

Submitted to:

Mr. GD Makkar

Associate Professor

Department of CA

**String**

Q1. Write a Python program to calculate the length of a string.

s="hello world"

print("the length of string:\"",s,"\" is :",len(s))

Output:



Q2. Write a Python program to count the number of characters (character frequency) in a string. Sample String : 'google.com'

s="hello world"

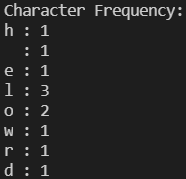
allchars=set(s)

print("Character Frequency:")

for char in allchars:

print(char,":",s.count(char))

Output:



Q3. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.

Sample String : 'abcdefghij'

Expected Result : 'abij'

Sample String : 'ab'

Expected Result : 'abab'

Sample String : ' a'

Expected Result : Empty String

s=input()

result=""

if(len(s)>=2):

result+=s[:2]

result+=s[-2:]

print(result)

else:

print("Empty String")

Output:





Q4. Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '$',except the first char itself.

Sample String : 'restart'

Expected Result : 'resta$t'

Code:

s=input()

result=''

result+=s[0]

for i in range(1,len(s)):

if s[0]==s[i]:

result+="$"

else:

result+=s[i]

print(result)



Q5. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

Sample String : 'abc', 'xyz'

Expected Result : 'xyc abz'

a="abc"

b="xyz"

result=""

result=b[:2]+a[2:]+' '+a[:2]+b[2:]

print(result)

Output:



Q6. Write a Python program to add 'ing' at the end of a given string

(length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.

Sample String : 'abc'

Expected Result : 'abcing'

Sample String : 'string'

Expected Result : 'stringly'

s=input()

if len(s)>=3:

if(s[-3:]=="ing"):

s=s+"ly"

else:

s=s+"ing"

print(s)

Output:



Q7. Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string.

Sample String : 'The lyrics is not that poor!'

'The lyrics is poor!'

Expected Result : 'The lyrics is good!'

'The lyrics is poor!'

s='the lyrics is not that poor'

start=s.find("not")

end=s.find("poor")

poor='poor'

if end>start:

s=s[:start]+"good"+s[end+len(poor):]

print(s)

Output:



Q8. Write a Python program that takes a list of words and returns the length of the longest one.

s=input("enter words with space in between:")

words=s.split()

longest\_len=0

for word in words:

if len(word)>longest\_len:

longest\_len=len(word)

print("the length of the longest word is:",longest\_len)

Output:



Q9. Write a Python program to remove the nth index character from a nonempty string.

s="Hello World"

n=6

s=s[:n]+s[n+1:]

print(s)

Output:



Q10. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically).

Sample Words : red, white, black, red, green, black

Expected Result : black, green, red, white

s=input()

words=s.split(',')

words=sorted(set(words))

print(",".join(words))

Output:



Q11. Write a Python program to reverses a string if it's length is a multiple of 4.

s="hello worlds"

if len(s)%4==0:

s=s[::-1]

print(s)

Output:



Q12. Write a Python function to convert a given string to all uppercase if it contains at least 2 uppercase characters in the first 4 characters.

s=input()

count=0

for i in range(4):

if s[i].isupper():

count+=1

if count>=2:

s=s.upper()

print(s)

Output:



Q13. Write a Python program to check whether a string starts with specified characters.

s=input()

char='a'

if s[0]==char:

print("Yes")

else:

print("No")

Output:



Q14. Write a Python program to create a Caesar encryption. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence.

s=input()

n=3

new\_s=''

for char in s:

new\_s+=chr(ord(char)-n)

print(new\_s)

Output:



Q15. Write a Python program to display a number in left, right and center aligned of width 10.

s="14"

num=10

print(s.ljust(num))

print(s.rjust(num))

print(s.center(num))

Output:



16. Write a Python program to reverse a string.

s='hello\_world'

print("original string:",s)

print("reversed string:",s[::-1])

Output:



17.Write a Python program to reverse words in a string.

s='hello world'

words=s.split()

for i in range(len(words)):

words[i]=words[i][::-1]

final\_str=" ".join(words)

print(final\_str)

Output:



18. Write a Python program to count repeated characters in a string.

Sample string:'thequickbrownfoxjumpsoverthelazydog' Expected output :

o 4

e 3

u 2

h 2

r 2

t 2

s="thequickbrownfoxjumpsoverthelazydog"

allchars=set(s)

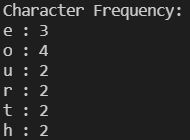
print("Character Frequency:")

for char in allchars:

if (s.count(char)>1):

print(char,":",s.count(char))

Output:



20. Write a Python program to count and display the vowels of a given text.

s="asdee34dfgdfg df00"

vowels=['a',0,'e',0,'i',0,'o',0,'u',0]

for i in range(0,len(vowels),2):

count=s.count(vowels[i])

vowels[i+1]=count

for i in range(0,len(vowels),2):

if vowels[i+1]>0:

print(vowels[i],":",vowels[i+1])

Output:



21. Write a Python program to find the first non-repeating character in given string.

s=input()

for i in range(len(s)):

if s.count(s[i])==1:

print("first non-repeating character:",s[i])

break

Output:



22. Write a Python program to find the first repeated word in a given string.

s=input()

all\_words=s.split()

for i in range(len(all\_words)):

if all\_words.count(all\_words[i])>1:

print("first repeated word:",all\_words[i])

break

Output:





23. Write a Python program to find the second most repeated word in a given string.

s=input()

all\_words=s.split()

words=list(set(all\_words))

count=[]

for word in words:

count.append(all\_words.count(word))

# print(words)

# print(count)

for i in range(len(count)):#bubble sort

for j in range(len(count)-i-1):

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

count[j]=count[j]+count[j+1]

count[j+1]=count[j]-count[j+1]

count[j]=count[j]-count[j+1]

temp=words[j]

words[j]=words[j+1]

words[j+1]=temp

# print(words)

# print(count)

print("second most used word is:",words[1])

Output:



24. Write a Python program to remove spaces from a given string.

s=input()

s="".join(s.split())

print(s)

Output:



25. Write a Python program to capitalize first and last letters of each word of a given string.

s=input()

words=s.split()

result=''

for i in range(len(words)):

result+=words[i][0].upper()+words[i][1:len(words[i])-1]

if len(words[i])!=1:

result+=words[i][-1].upper()

if i!=len(words)-1:

result+=' '

print(result)

Output:



26. Write a Python program to remove duplicate characters of a given string.

s='aaaasssssssdaggggdddfdafsafsssd'

out=''

out+=s[0]

for i in range(1,len(s)):

if out[-1]!=s[i]:

out+=s[i]

print("Original string:",s)

print("Processed string:",out)

Output:



27. Write a Python program to create two strings from a given string. Create the first string using those character which occurs only once and create the second string which consists of multi-time occurring characters in the said string.

s=input()

once=''

repeating=''

for char in s:

if s.count(char)==1:

once+=char

else:

if char not in repeating:

repeating+=char

print("character which occured once:",once)

print("character which occured multiple times:",repeating)

Output:



28. Write a Python program to create a string from two given strings concatenating uncommon characters of the said strings.

a="asdfe"

b="qwasdes"

both=a+b

bothchars=set(both)

c=""

for char in bothchars:

if both.count(char)==1:

c+=char

print(c)

Output:



29. Write a Python program to count Uppercase, Lowercase, special character and numeric values in a given string.

s="23efasdf77E$^"

u\_c=0

l\_c=0

n\_c=0

s\_c=0

for char in s:

if char.isupper():

u\_c+=1

elif char.islower():

l\_c+=1

elif char.isnumeric():

n\_c+=1

else:

s\_c+=1

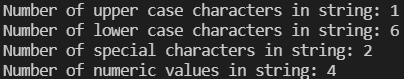
print("Number of upper case characters in string:",u\_c)

print("Number of lower case characters in string:",l\_c)

print("Number of special characters in string:",s\_c)

print("Number of numeric values in string:",n\_c)

Output:



30. Write a Python program to swap cases of a given String.

s=input()

print(s.swapcase())

Output:



**List**

1. Write a Python program to get the largest and second largest number from a list.

l1=[12,56,1,3,6,7,23,42,6]

l1.sort(reverse=True)

print('largest number:',l1[0])

print('Second largest number:',l1[1])

Output:



1. Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.

Sample List : ['abc', 'xyz', 'aba', '1221']

Expected Result : 2

s1=['abc', 'xyz', 'aba', '1221']

count=0

for s in s1:

if s[0]==s[-1] and len(s)>=2:

count+=1

print(count)

Output:



1. Write a Python program to remove duplicates from a list.

l1=["ss","tyu","ss","yu","tyu"]

l2=[]

l2.append(l1[0])

for i in l1[1:]:

if i not in l2:

l2.append(i)

print(l2)

Output:



1. Write a Python program to get the difference between the two lists.

l1=[2,4,6,1]

l2=[1,5,2,5,6,7,8]

out=[]

for i in l1+l2:

if not (i in l1 and i in l2):

# if (i in l1 or i in l2) and not (i in l1 and i in l2):

out.append(i)

print('list 1:',l1)

print('list 2:',l2)

print('difference:',out)

Output:



1. Write a Python program to convert a list of characters into a string.

l1=['a','n','u','j']

print("list:",l1)

out=''.join(l1)

print(out)

Output:



1. Write a Python program to find the index of an item in a specified list.

l1=['mango','apple','banana','grapes']

to\_find=input("enter item to find index of: ")

idx=l1.index(to\_find)

print("the index of item \'"+to\_find+"\' is :",idx)

Output:



1. Write a Python program to append a list to the second list.

l1=['he','llo ']

l2=['w','o','r','l','d']

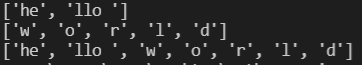
print(l1)

print(l2)

l1.extend(l2)

print(l1)

Output:



1. Write a Python program to get unique values from a list.

l1=[1,2,7,7,5,2,5,6,7,8]

out=[]

for i in range(len(l1)):

repeat=False

for j in range(len(l1)):

if l1[i]==l1[j] and i!=j:

repeat=True

break

if not repeat:

out.append(l1[i])

print('list:',l1)

print("unique elements:",out)

Output:



1. Write a Python program to get the frequency of the elements in a list.

l1=[1,2,7,7,5,2,5,6,7,8]

out=[]

for i in l1:

if out.count([i,l1.count(i)])==0:

out.append([i,l1.count(i)])

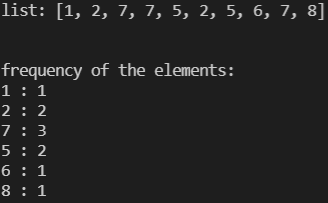
print('list:',l1,'\n\n')

print('frequency of the elements:')

for item in out:

print(item[0],":",item[1])

Output:



11.Write a Python program to find common items from two lists.

l1=[2,4,6,1]

l2=[1,5,2,5,6,7,8]

out=[]

for i in l1+l2:

if (i in l1 and i in l2) and i not in out:

# if (i in l1 or i in l2) and not (i in l1 and i in l2):

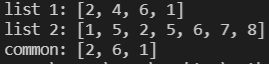
out.append(i)

print('list 1:',l1)

print('list 2:',l2)

print('common:',out)

Output:



12. Write a Python program to convert a list of multiple integers into a single integer.

Sample list: [11, 33, 50]

Expected Output: 113350

l1=[2,4,6,1]

l1=list(map(str,l1))

out=int("".join(l1))

print("list:",l1)

print("value:",out)

Output:



13. Write a Python program to replace the last element in a list with another list.

Sample data : [1, 3, 5, 7, 9, 10], [2, 4, 6, 8]

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

l1=[1, 3, 5, 7, 9, 10]

l2=[2,4,6,8]

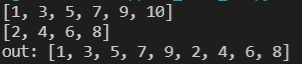
print(l1)

print(l2)

l1=l1[:-1]+l2

print('out:',l1)

Output:



14. Write a Python program to find the list in a list of lists whose sum of elements is the highest.

Sample lists: [1,2,3], [4,5,6], [10,11,12], [7,8,9]

Expected Output: [10, 11, 12]

l1=[[1,2,4],[2,1,6],[2,7,4],[1,4,2]]

highest\_sum=[]

for i in l1:

if sum(i)>sum(highest\_sum):

highest\_sum=i

print('list:',l1)

print('list of lists whos sum is highest:',highest\_sum)

Output:



15. Write a Python program to remove duplicates from a list of lists.

Sample list : [[10, 20], [40], [30, 56, 25], [10, 20], [33],[40]]

New List : [[10, 20], [30, 56, 25], [33], [40]]

l1=[[10, 20], [40], [30, 56, 25], [10, 20], [33],[40]]

l2=[]

l2.append(l1[0])

for i in l1[1:]:

if i not in l2:

l2.append(i)

print(l2)

Output:



16.Write a Python program to flatten a given nested list structure.

Original list: [0, 10, [20, 30], 40, 50, [60, 70, 80], [90, 100,110, 120]]

Flatten list:[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]

l1=[0, 10, [20, 30], 40, 50, [60, 70, 80], [90, 100,110, 120]]

l2=[]

for i in l1:

if type(i)==list:

l2.extend(i)

else:

l2.append(i)

print(l2)

Output:



17. Write a Python program to remove consecutive duplicates of a given list.

Original list:

[0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 8, 9, 4, 4]

After removing consecutive duplicates:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 4]

l1=[0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 8, 9, 4, 4]

l2=[l1[0]]

for i in l1[1:]:

if i!=l2[-1]:

l2.append(i)

print("original List:",l1)

print("Final List:",l2)

Output:



18. Write a Python program to remove the K'th element from a given list, print the new list.

l1=[1,2,4,5,1,2,423]

print('original list:',l1)

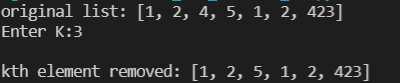
k=int(input("Enter K:"))-1

print()

l2=l1[:k]+l1[k+1:]

print('kth element removed:',l2)

Output:



**Tuple**

1. Write a Python program to calculate the average value of

the numbers in a given tuple of tuples.

Original Tuple:

((10, 10, 10, 12), (30, 45, 56, 45), (81, 80, 39, 32), (1, 2, 3,4))

Average value of the numbers of the said tuple of tuples:

[30.5, 34.25, 27.0, 23.25]

Original Tuple:

((1, 1, -5), (30, -15, 56), (81, -60, -39), (-10, 2, 3))

Average value of the numbers of the said tuple of tuples:

[25.5, -18.0, 3.75]

t1=((10, 10, 10, 12), (30, 45, 56, 45), (81, 80, 39, 32), (1, 2, 3,4))

avg\_list=[]

for t in t1:

avg\_list.append(sum(t)/len(t))

print(t1)

print(avg\_list)

Output:



2. Write a Python program to convert a given tuple of positive integers into an integer.

Original tuple:

(1, 2, 3)

Convert the said tuple of positive integers into an integer:

123

Original tuple:

(10, 20, 40, 5, 70)

Convert the said tuple of positive integers into an integer:

102040570

t1=(10, 10, 10, 12)

num=''.join(tuple(map(str,t1)))

num=int(num)

print('t1:',t1)

print('num:',num)

Output:



3. Write a Python program to check if a specified element

presents in a tuple of tuples.

Original list:

(('Red', 'White', 'Blue'), ('Green', 'Pink', 'Purple'), ('Orange',

'Yellow', 'Lime'))

Check if White present in said tuple of tuples!

True

Check if White present in said tuple of tuples!

True

Check if Olive present in said tuple of tuples!

False

t1=(('Red', 'White', 'Blue'), ('Green', 'Pink', 'Purple'), ('Orange','Yellow', 'Lime'))

val=input('enter element to search:')

print()

for t in t1:

if t.count(val)>0:

print(val,' is present is tuple of tuples')

break

else:

print(val,' is not present is tuple of tuples')

Output:



4. Write a Python program to convert a given list of tuples to a list of lists.

Original list of tuples: [(1, 2), (2, 3), (3, 4)]

Convert the said list of tuples to a list of lists: [[1, 2], [2, 3],

[3, 4]]

Original list of tuples: [(1, 2), (2, 3, 5), (3, 4), (2, 3, 4, 2)]

Convert the said list of tuples to a list of lists: [[1, 2], [2, 3,

5], [3, 4], [2, 3, 4, 2]]

l1=[(1, 2), (2, 3), (3, 4)]

print("Original list:",l1)

for i in range(len(l1)):

l1[i]=list(l1[i])

print("Final list:",l1)

Output:



5. Swap the following two tuples

t1=(1,2)

t2=(3,4)

t1=(1,2)

t2=(3,4)

print('Originally:')

print('t1:',t1)

print('t2:',t2)

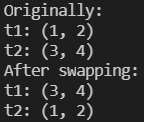
print('After swapping:')

t1,t2=t2,t1

print('t1:',t1)

print('t2:',t2)

Output:



6. Sort a tuple of tuples by 2nd item

tuple1 = (('a', 23),('b', 37),('c', 11), ('d',29))

t1 = (('a', 23),('b', 37),('c', 11), ('d',29))

print('Original t1:',t1)

print('Sorted t1:',tuple(sorted(t1,key=lambda x:x[1])))

Output:



7. Check if all items in the following tuple are the same.

tuple1 = (45, 45, 45, 45)

Expected output:

True

t1 = (45,45,45,45,45)

print('t1:',t1)

print('all are same:',t1.count(t1[0])==len(t1))

Output:



**Sets**

1. Write a Python program to add member(s) in a set and iterate over it.

s1=set()

# print(type(s1))

n=int(input('number of elements to add to set:'))

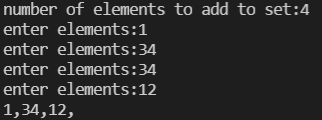
for i in range(n):

s1.add(int(input('enter elements:')))

for i in s1:

print(i,end=',')

Output:



2. Write a Python program to remove an item from a set if it is present in the set.

s1={1,3,5,2,6,}

print('set:',s1)

element=int(input('element to remove:'))

s1.remove(element)

print("final set:",s1)

Output:



3. Write a Python program to apply union, intersection, difference,symmetric difference operators on sets.

s1={1,3,5,2,6}

s2={2,5,3,4,5,17,3}

print("set 1:",s1)

print("set 2:",s2)

print("Union:",s1.union(s2))

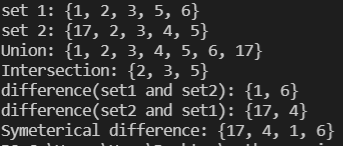
print("Intersection:",s1.intersection(s2))

print('difference(set1 and set2):',s1.difference(s2))

print('difference(set2 and set1):',s2.difference(s1))

print('Symeterical difference:',s1.symmetric\_difference(s2))

Output:



4. Write a Python program to check if a set is a subset of another set.

s1={2,5,3,4,5,17,3}

s2={1,3,5,2,6}

s3={3,5,2}

print('s1:',s1)

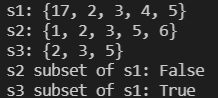
print('s2:',s2)

print('s3:',s3)

print('s2 subset of s1:',s2.issubset(s1))

print('s3 subset of s1:',s3.issubset(s1))

Output:



5. Write a Python program to clear a set.

s1={2,5,3,4,5,17,3}

print('s1:',s1)

s1.clear()

print('s1 after being cleared:',s1)

Output:



6. Write a Python program to check if two given sets have no

elements in common.

s1={2,5,3,4,5,17,3}

s2={1,3,5,2,6}

print('s1:',s1)

print('s2:',s2)

if len(s1.intersection(s2))==0:

print("no common elements found")

else:

print("common elements found")

Output:



7. Write a Python program to remove the intersection of a 2nd set from the 1st set.

s1={2,5,3,4,5,17,3}

s2={1,3,5,2,6}

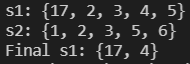
print('s1:',s1)

print('s2:',s2)

s1.difference\_update(s2)

print('Final s1:',s1)

Output:



**Dictionary**

1. Write a Python script to sort (ascending and descending) a dictionary by value.

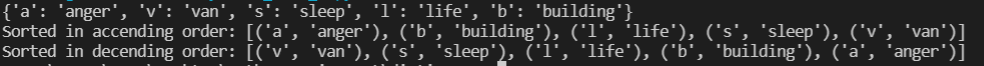
d1={'a':'anger','v':'van','s':'sleep','l':'life','b':'building'}

print(d1)

print("Sorted in accending order:",sorted(d1.items(),key=(lambda x:x[1]) ) )

print("Sorted in decending order:",sorted(d1.items(),key=(lambda x:x[1]) ,reverse=True) )

Output:



2. Write a Python script to add a key to a dictionary.

Sample Dictionary : {0: 10, 1: 20}

Expected Result : {0: 10, 1: 20, 2: 30}

d1={1:10,2:20}

print("originally:",d1)

d1[3]=30

print("final:",d1)

Output:



3. Write a Python script to concatenate following dictionaries to create a new one.

Sample Dictionary :

dic1={1:10, 2:20}

dic2={3:30, 4:40}

dic3={5:50,6:60}

Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

d1={1:10, 2:20}

d2={3:30, 4:40}

d3={5:50,6:60}

print('d1:',d1)

print('d2:',d2)

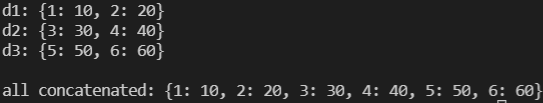
print('d3:',d3)

d1.update(d2)

d1.update(d3)

print('\nall concatenated:',d1)

Output:



4. Write a Python script to check whether a given key already exists in a dictionary.

d1={'1':10,'2':20,'4':40,'5':50}

k=input("enter key to search for:")

if list(d1.keys()).count(k)==0:

print('\"',k,"\" key not found in dictionary")

else:

print('\"',k,"\" key found in dictionary")

Output:



5. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x\*x).

Sample Dictionary ( n = 5) :

Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

n=int(input('n:'))

d1={i:i\*i for i in range(1,n+1)}

print(d1)

Output:



6. Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys.

Sample Dictionary

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11:121, 12: 144, 13: 169, 14: 196, 15: 225}

d1={i:i\*i for i in range(1,15+1)}

print(d1)

Output:



7. Write a Python script to merge two Python dictionaries.

d1={1:10, 2:20}

d2={3:30, 4:40}

print('d1:',d1)

print('d2:',d2)

d1.update(d2)

print("merged dict:",d1)

Output:



9. Write a Python program to sort a dictionary by key.

d1={'a':'anger','v':'van','s':'sleep','l':'life','b':'building'}

print(d1)

print(sorted(d1.items(),key=(lambda x:x[0]) ) )

Output:



10. Write a Python program to remove duplicate values from Dictionary.

d1={'a':'anger','v':'van','s':'sleep','an':'abroad','l':'life','b':'building'}

print(d1)

Output:



11. Write a Python program to combine two dictionary adding values for common keys.

d1 = {'a': 100, 'b': 200, 'c':300}

d2 = {'a': 300, 'b': 200, 'd':400}

Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})

d1 = {'a': 100, 'b': 200, 'c':300}

d2 = {'a': 300, 'b': 200, 'd':400}

print('d1:',d1)

print('d1:',d2)

for key in d2:

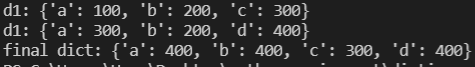
if key not in d1.keys():

d1[key]=0

d1[key]+=d2[key]

print('final dict:',d1)

Output:



12.Write a Python program to find the highest 3 values in a dictionary.

d1={'a':'anger','v':'van','s':'sleep','an':'abroad','l':'life','b':'building'}

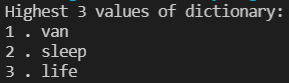
sorted\_values=sorted(list(d1.values()),reverse=True)

print('Highest 3 values of dictionary:')

for i in range(3):

print(i+1,'.',sorted\_values[i])

Output:



13. Write a Python program to get the top three items in a shop.

Sample data: {'item1': 45.50, 'item2':35, 'item3':41.30,'item4':55,'item5': 24}

Expected Output:

item4 55

item1 45.5

item3 41.3

d1={'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55,'item5': 24}

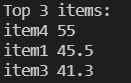
sorted\_dict=sorted(list(d1.items()),key=lambda x:x[1],reverse=True)

print("Top 3 items:")

for item in sorted\_dict[:3]:

print(item[0],item[1])

Output:



14. Write a Python program to filter a dictionary based on values.

Original Dictionary:

{'Cierra Vega': 175, 'Alden Cantrell': 180, 'Kierra Gentry': 165, 'Pierre Cox': 190}

Marks greater than 170:

{'Cierra Vega': 175, 'Alden Cantrell': 180, 'Pierre Cox': 190}

d1={'Cierra Vega': 175, 'Alden Cantrell': 180, 'Kierra Gentry': 165, 'PierreCox': 190}

print('d1:',d1,'\n')

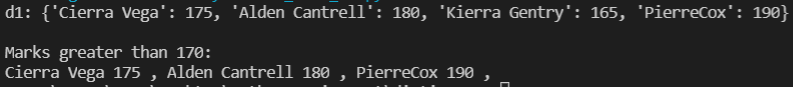
val=list(filter(lambda x:d1[x]>170,d1))

print('Marks greater than 170:')

for v in val:

print(v,d1[v],', ',end='')

Output:



**Functions**

1. Write a Python function to sum all the numbers in a list. Sample List : (8, 2, 3, 0, 7)

Expected Output : 20

def sum\_func(list1):

out=sum(list1)

return out

l1=[8,2,5,4,1,4,2]

print('l1:',l1)

print('sum of l1:',sum\_func(l1))

Output:



2. Write a Python program to reverse a string.

Sample String : "1234abcd"

Expected Output : "dcba4321"

def rev(str1):

return str1[::-1]

s1='12345abcd'

print('s1:',s1)

print('Reverse of s1:',rev(s1))

Output:



3. Write a Python function to calculate the factorial of a number (a non- negative integer). The function accepts the number as an argument.

def fac(num):

out=num

for i in range(2,num):

out\*=i

return out

print('factorial of 3:',fac(3))

print('factorial of 4:',fac(4))

Output:



4. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters. Sample String : 'The quick Brow Fox' Expected Output :

No. of Upper case characters : 3

No. of Lower case Characters : 12

def count\_upper\_lower(str1):

lower=0

upper=0

for ch in str1:

if ch.islower():

lower+=1

elif ch.isupper():

upper+=1

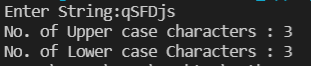
print('No. of Upper case characters :',upper)

print('No. of Lower case Characters :',lower)

s1=input("Enter String:")

count\_upper\_lower(s1)

Output:



5. Write a Python function to check whether a number is perfect or not.

Note: In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum). Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).

Example : The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: (1 + 2 + 3 + 6) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.

def perfect\_number(num):

sum=0

for i in range(1,num):

if num%i==0:

sum+=i

if sum==num:

return True

else:

return False

n=int(input('Enter a Number:'))

print(n,'is perfect number:',perfect\_number(n))

Output:



6. Write a Python function to check whether a string is a pangram or not.

**Note** : Pangrams are words or sentences containing every letter of the alphabet at least once.

For example : "The quick brown fox jumps over the lazy dog”

def pangram(str1):

str1=str1.lower()

allchars='abcdefghijklmnopqrstuvwxyz'

pangram=True

for char in allchars:

if char not in str1:

pangram=False

break

return pangram

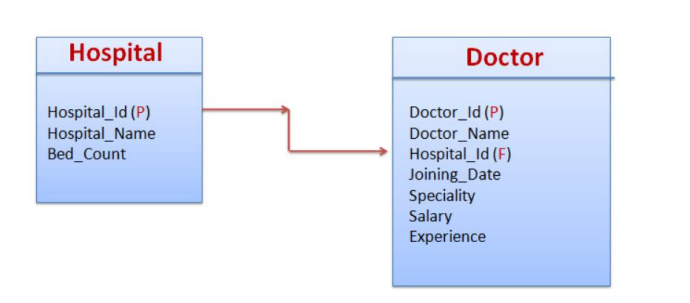
s=input("Enter String:")

print(s,'is pangram:',pangram(s))

Output:



**SQLite**



1.Create tables Hospital and Doctor.

import sqlite3

con=sqlite3.connect('database.db')

cur=con.cursor()

cur.execute('drop table Hospital')

cur.execute('drop table Doctor')

cur.execute('create table Hospital(Hospital\_id char primary key,Hospital\_Name char,Bed\_Count int);')

cur.execute('create table Doctor(Doctor\_id char primary key,Doctor\_Name char,Hospital\_id char,Joining\_Date date,Speciality char,Salary int,Experience int,FOREIGN KEY(Hospital\_id) REFERENCES Hospital(Hospital\_id));')

2. Insert records in tables Hospital and Doctor by taking input from user.

import sqlite3

from datetime import date

con=sqlite3.connect("database.db")

cur=con.cursor()

cur.execute("PRAGMA foreign\_keys = 1")

ch=input('Enter h or d for inserting hospital or doctor details:')

if ch.lower()=='h':

h\_id=input("Hospital Id:")

h\_name=input("Hospital Name:")

b\_count=input("Bed Count:")

con.execute(f'insert into Hospital values(\'{h\_id}\',\'{h\_name}\',\'{b\_count}\')')

elif ch.lower()=='d':

d\_id=input('Doctor Id:')

d\_name=input('Doctor Name:')

h\_id=input('Hospital Id:')

j\_date=input('Joining Date(DD-MM-YYYY):').split('-')

j\_date=list(map(int,j\_date))

j\_date=date(j\_date[-1],j\_date[-2],j\_date[-3])

print(j\_date)

spec=input('Speciality:')

salary=input('Salary:')

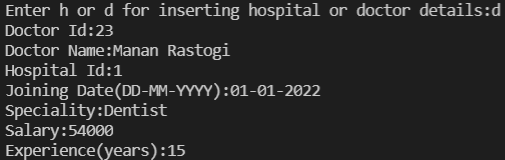
experience=input('Experience(years):')

con.execute(f'insert into Doctor values(\'{d\_id}\',\'{d\_name}\',\'{h\_id}\',\'{j\_date}\',\'{spec}\',\'{salary}\',\'{experience}\')')

con.commit()

Output:





3. Fetch Hospital and Doctor Information using hospital Id and doctor Id

Implement the functionality to read the details of a given doctor from the doctor table and Hospital from the hospital table. i.e., read records from Hospital and Doctor Table as per given hospital Id and Doctor Id.

import sqlite3

con=sqlite3.connect('database.db')

cur=con.cursor()

ch=input("Enter h or d for hospital or doctor:")

if ch=='h':

h\_id=input("Enter Hospital Id:")

cur.execute(f"Select \* from Hospital where Hospital\_id=\'{h\_id}\'")

# cur.execute("Select \* from Hospital")

data=cur.fetchall()

names = [description[0] for description in cur.description]

for name in names:

print("\t",name, end='',sep='')

print()

for i,rows in enumerate(data):

print((i+1),'. ',sep='',end='')

print("\t",end='')

for j in range(len(rows)):

print('\t',rows[j],end=' ')

elif ch=='d':

d\_id=input("Enter Doctor Id:")

cur.execute(f"Select \* from Doctor where Doctor\_id=\'{d\_id}\'")

data=cur.fetchall()

names = [description[0] for description in cur.description]

for name in names:

print("\t",name, end='',sep='')

print()

for i,rows in enumerate(data):

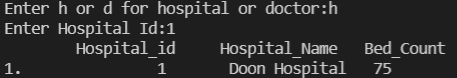
print((i+1),'. ',sep='',end='')

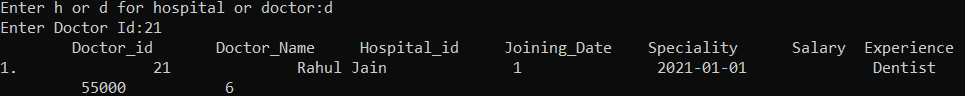
# print("\t",end='')

for j in range(len(rows)):

print('\t\t',rows[j],end=' ')

Output:





4. Get the list of doctors as per the given specialty and salary

Fetch all doctors whos salary higher than the input amount and specialty is the same as the input specialty.

import sqlite3

con=sqlite3.connect('database.db')

cur=con.cursor()

min\_sal=input("Enter minimum salary:")

spec=input("Enter Speciality:")

cur.execute(f"Select Doctor\_name from Doctor where salary>\'{min\_sal}\' and Speciality=\'{spec}\'")

# cur.execute("Select \* from Hospital")

data=cur.fetchall()

names = [description[0] for description in cur.description]

for name in names:

print("\t",name, end='',sep='')

print()

for i,rows in enumerate(data):

print((i+1),'. ',sep='',end='')

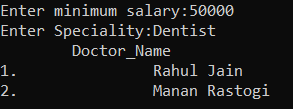
print("\t",end='')

for j in range(len(rows)):

print('\t',rows[j],end=' ')

print()

Output:



5. Get a list of doctors from a given hospital

Implement the functionality to fetch all the doctors as per the given Hospital Id. You must display the hospital name of a doctor.

import sqlite3

con=sqlite3.connect('database.db')

cur=con.cursor()

h\_id=input("Enter Hospital Id:")

cur.execute(f"Select Doctor\_name,Hospital\_name from Hospital inner join Doctor ON Hospital.Hospital\_id=Doctor.Hospital\_id and Hospital.Hospital\_id=\'{h\_id}\'")

# cur.execute("Select \* from Hospital")

data=cur.fetchall()

names = [description[0] for description in cur.description]

for name in names:

print("\t",name, end='',sep='')

print()

for i,rows in enumerate(data):

print((i+1),'. ',sep='',end='')

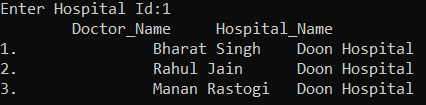
print("\t",end='')

for j in range(len(rows)):

print('\t',rows[j],end=' ')

print()

Output:





6. Update doctor experience in years

The value of the experience column for each doctor is null. Implement the functionality to update the experience of a given doctors in years.

import sqlite3

con=sqlite3.connect('database.db')

cur=con.cursor()

d\_id=input("Enter Doctor Id:")

exp=input("Enter Experience(in years):")

cur.execute(f"Update Doctor set Experience=\'{exp}\' where Doctor\_id=\'{d\_id}\'")

print("Updated Successfully!")

con.commit()

Output:

