Python Assignment

Name = jay geda

Section = ‘U’

University Roll No = 2215000844

Submitted To = Gurpreet Kaur Mam Course = B.tech C.S.

1. Python Program for n-th Fibonacci number

|  |
| --- |
| n=int(input("Enter the Number")) f=0 s=1 if n==0: print(f)  elif n==1: print(f,s)  else:  print(f,s,end=' ') for i in range(2,n+1):  temp=f+s f=s s=temp print(temp,end=' ') |

1. Python Program for How to check if a given number is Fibonacci number?

|  |
| --- |
| n=int(input("Enter the number: ")) n1=0 n2=1 n3=1 if n==0 or n==1:  print("Yes")  else:  while n1<n: n1=n2+n3 n3=n2 n2=n1 if n1==n: print("Yes")  else: |

print("No")

1. Python Program for n\’th multiple of a number in Fibonacci Series

|  |
| --- |
| n1=int(input('Enter the number:')) n2=int(input('Enter the nth:')) f1 = 0 f2 = 1 i =2 while i!=0: f3 = f1 + f2 f1 = f2 f2 = f3 if f2%n1 == 0:  print(n2\*i)  i+=1 |

1. Program to print ASCII Value of a character

x=(input('Enter the character'))

print("The ASCII Value of Character x is",ord(x))

1. Python Program for Sum of squares of first n natural numbers

x=int(input('Enter the Number')) for i in range(1,x+1):

print(i\*\*2,end=' ')

1. Write a Python program to swap two numbers using bitwise operator.

a=int(input("Enter the number")) b=int(input("Enter the number")) x=a^b b=x^b a=x^a

print(a,b)

1. Write a Python program to check whether a character is alphabet or not.

|  |
| --- |
| x=input() for i in range(65,91):  if(chr(i)==x) or (chr(i+32)==x):  print("Alphabet") break  else:  print("Not a Albahbet") |

1. Write a Python program to input any alphabet and check whether it is vowel or consonant.

|  |
| --- |
| x=input() if(x==chr(97) or x==chr(102) or x==chr(108) or x==chr(112) or x==chr(118)):  print("Vowel")  else:  print("Consonant") |

1. Write a Python program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer. Calculate percentage and grade according to following:

Percentage >= 90% : Grade A

Percentage >= 80% : Grade B

Percentage >= 70% : Grade C

Percentage >= 60% : Grade D

Percentage >= 40% : Grade E

Percentage < 40% : Grade F

|  |
| --- |
| x=int(input("Enter the marks of physics")) y=int(input("Enter the marks of Chemistry")) z=int(input("Enter the marks of mathematics")) w=int(input("Enter the marks of computer")) |
| u=int(input("Enter the marks of biology")) |
| mark=(x+y+z+u+w)/5 if(mark>=90 and mark<100): print("Grade A")  elif(mark>=80 and mark<90): print("Grade B")  elif(mark>=70 and mark<80): print("Grade C")  elif(mark>=60 and mark<70): print("Grade D")  elif(mark>=50 and mark<60): print("Grade E")  elif(mark>=40 and mark<50):  print("Grade F")  else:  print("Fail") |

1. Write a Python program to input basic salary of an employee and calculate its Gross salary according to following:

Basic Salary <= 10000 : HRA = 20%, DA = 80%

Basic Salary <= 20000 : HRA = 25%, DA = 90%

Basic Salary > 20000 : HRA = 30%, DA = 95%

|  |
| --- |
| x=int(input("Enter the Salary of user")) if(x>=20000): hra=x\*0.2 da=x\*0.8  elif(x>=10000 and x<=20000):  hra=x\*0.25 da=x\*0.9  else:  hra=x\*0.3 da=x\*0.95  gross=x+hra+da print("The total gross of User is ",gross) |

1. Write a Python program to input electricity unit charges and calculate total electricity bill according to the given condition:

For first 50 units Rs. 0.50/unit

For next 100 units Rs. 0.75/unit

For next 100 units Rs. 1.20/unit

For unit above 250 Rs. 1.50/unit

An additional surcharge of 20% is added to the bill

|  |
| --- |
| unit=int(input()) if(unit<=50):  bill=unit\*0.5  elif(unit>=50 and unit<=150): bill=(unit-50)\*0.75 + 50\*0.5  elif(unit>=150 and unit<=250):  bill=50\*0.5 + (unit-150)\*1.2 + (unit-100)\*0.75  else:  bill=50\*0.5 + (unit-150)\*1.2 + (unit-100)\*0.75 +  (unit-250)\*1.50 bill = bill + bill\*0.2 print(bill) |

1. Write a Python program to print all alphabets from a to z. – using while Loop

x=1 while(x<27): print(chr(x+96),end=' ') x=x+1

1. Write a Python program to find first and last digit of a number.

|  |
| --- |
| x=input('Enter the Number') l=len(x) print("The first digit of ",x,'is ',(int(x)//(10\*\*(l-  1)))) print("The last digit of ",x,"is ",int(x)%10) |

1. Write a Python program to calculate sum of digits of a number.

num=int(input("Enter the number")) sum=0 while(num!=0): sum=sum+(num)%10 num=num//10

print(sum)

1. Write a Python program to calculate product of digits of a number.

num=int(input("Enter the number")) sum=1 while(num!=0): sum=sum\*(num)%10 num=num//10

print(sum)

1. Write a Python program to enter a number and print its reverse.

num=int(input("Enter the number")) l=len(str(num)) sum=0 while(num!=0): sum=sum+((num)%10)\*(10\*\*(l-1)) num=num//10 l=l-1 print(sum)

1. Write a Python program to check whether a number is palindrome or not.

|  |
| --- |
| num=int(input("Enter the number")) l=len(str(num)) check=num sum=0 while(num!=0): sum=sum+((num)%10)\*(10\*\*(l-1)) num=num//10 l=l-1  if(check==sum):  print("Palindrome")  else:  print("Not a palindrome") |

1. Write a Python program to find all factors of a number.

|  |
| --- |
| x=int(input("Enter the number")) print("The factors are ",end='') for i in range(1,x+1):  if(x%i==0):  print(i,end=' ') |

1. Write a Python program to calculate factorial of a number.

|  |
| --- |
| x=int(input("Enter the number")) print("The factorial of",x,"is ",end='') rem=1 for i in range(2,x+1):  rem=rem\*i  print(rem) |

1. Write a Python program to find HCF (GCD) of two numbers.

|  |
| --- |
| x=int(input("Enter the first number")) y=int(input("Enter the second Number")) if(x>y):  for i in range(1,y+1):  if(x%i==0 and y%i==0):  rem=i  else:  for i in range(1,x+1):  if(x%i==0 and y%i==0):  rem=i  print(rem) |

1. Write a Python program to find LCM of two numbers.

|  |
| --- |
| x=int(input("Enter the first number")) y=int(input("Enter the second Number")) rem=1 if(x>y):  for i in range(2,x+1):  if(x%i==0 or y%i==0):  rem=rem\*i  else: |
| for i in range(2,y+1):  if(x%i==0 or y%i==0):  rem=rem\*i  print(rem) |

1. Write a Python program to check whether a number is Prime number or not.

|  |
| --- |
| x=int(input("Enter the Number")) for i in range(2,x):  if(x%i==0): print("Not a prime number") break  else:  print("Prime Number") |

1. Write a Python program to print all Prime numbers between 1 to n.

|  |
| --- |
| x=int(input("Enter the Number")) for j in range(2,x+1):  for i in range(2,j):  if(j%i==0):  break  else:  print(j,end=' ') |

1. Write a Python program to find sum of all prime numbers between 1 to n.

x=int(input("Enter the Number"))

|  |
| --- |
| sum=0 for j in range(2,x+1):  for i in range(2,j):  if(j%i==0):  break  else: sum=sum+j  print(sum) |

1. Write a Python program to find all prime factors of a number

|  |
| --- |
| x=int(input()) z=x y=2 while(x!=1):  if(x%y==0): print(y,end=' ') x=x//y  else: y=y+1 |

1. Write a Python program to check whether a number is Armstrong number or not.

|  |
| --- |
| x=int(input("Enter the Number")) l=len(str(x)) z=x sum=0 for i in range(0,l):  rem=x%10 |
| sum=sum+rem\*\*l x=x//10 if(z==sum):  print("The Number is Armstrong")  else:  print("Not a Armstrong") |

1. Write a Python program to print all Armstrong numbers between 1 to n

x=int(input("Enter the Number")) for j in range(153,x): sum=0 l=len(str(j)) z=j k=j for i in range(0,l): rem=k%10 sum=sum+rem\*\*l k=k//10 if(z==sum):

print(j)

1. Write a Python program to check whether a number is Perfect number or not.

|  |
| --- |
| x=int(input()) sum=0 y=x for i in range(1,x):  if(x%i==0): sum=sum+i  if(y==sum): |
| print("Perfect Number")  else:  print("Not a perfect Number") |

1. Write a Python program to check whether a

number is Strong number or not (Also known as Robinson number/ Krishnamurthy Number / Peterson number.)

|  |
| --- |
| x=int(input("Enter the number")) sum=0 l=len(str(x)) z=x for j in range(0,l):  mul=1 rem=x%10 for i in range(1,rem+1):  mul=mul\*i  sum=sum+mul x=x//10 if(z==sum):  print("Strong Number")  else:  print("Not a Strong Number") |

1. Python program to check whether the string is Symmetrical or Palindrome

string = input() half = int(len(string) / 2) first\_str = string[:half] second\_str = string[half:] if first\_str == second\_str:

print(string, 'string is symmetrical') else:

print(string, 'string is not symmetrical')

if first\_str == second\_str[::-1]:

print(string, 'string is palindrome') else: print(string, 'string is not palindrome')

1. Reverse words in a given String in Python

|  |
| --- |
| string = "geeks quiz practice code" s = string.split()[::-1] l = [] for i in s:  l.append(i) print(" ".join(l)) |

1. Ways to remove i’th character from string in Python

|  |
| --- |
| x=input("Enter the string") l=len(x) nx='' n=int(input("Enter the position of which you want to remove the element")) for i in range (0,l):  if(i!=n-1):  nx=nx+x[i]  print(nx) |

1. Python program to Check if a Substring is Present in a Given String

|  |
| --- |
| x=input("Enter the string ") z=x.split() l=len(z) f=input("Enter the find String") for i in range(0,l):  if(z[i]==f): print("The Element is found") break  else:  print("Element is not found") |

1. Python program to count words frequency in String Shorthands

x=input("Enter the striing") z=x.split() print(len(z))

1. Python program to convert snake case to pascal case

|  |
| --- |
| test\_str = 'geeksforgeeks\_is\_best' print("The original string is : " + test\_str)  res = test\_str.replace("\_", " ").title().replace(" ",  "") print("The String after changing case : " + str(res)) |

1. Python program to print even length words in a string

|  |
| --- |
| x=input("Enter the String")  z=x.split() l=len(z) for i in range(0,l):  if(i%2!=0): print(z[i]) |

1. Python program to accept the strings which contains all vowels

|  |
| --- |
| x=input() if('a' in x and 'e' in x and 'i' in x and 'o' in x and 'u' in x): print("All vowels are present")  else:  print("Not present") |

1. Python program to count the Number of matching characters in a pair of string

x=input("Enter the first string") y=input("Enter the second string") x1=set(x) y1=set(y) match= (x1 & y1) print(len(match))

1. Remove all duplicates from a given string in Python

|  |
| --- |
| string="geeksforgeeks" p="" for char in string:  if char not in p:  p=p+char  print(p) k=list("geeksforgeeks") |

1. Python programs to count Least Frequent Character in String

|  |
| --- |
| string= "hgfjiufguij" print(string) char\_freq={} for i in string:  if i in char\_freq:  char\_freq[i]=char\_freq[i]+1  else:  char\_freq[i] = 1  result= min(char\_freq, key = char\_freq.get) print("Least frequent character: ",result) |

1. Python programs to count maximum frequency character in String

test\_str = "GeeksforGeeks" print ("The original string is : " + test\_str) all\_freq = {} for i in test\_str:

|  |
| --- |
| if i in all\_freq:  all\_freq[i] += 1  else:  all\_freq[i] = 1  res = max(all\_freq, key = all\_freq.get) print ("The maximum of all characters in GeeksforGeeks is : " + str(res)) |

45. Python program to split and join a string

s = 'Geeks for Geeks' print(s.split(" ")) print("-".join(s.split()))

46. Python program to find uncommon words from two Strings

|  |
| --- |
| 1. = "Geeks for Geeks" 2. = "Learning from Geeks for Geeks"   A=A.split() B=B.split() x=[] for i in A:  if i not in B:  x.append(i)  for i in B:  if i not in A:  x.append(i)  x=list(set(x)) print(x) |

1. String slicing in Python to rotate a string

|  |
| --- |
| input = 'GeeksforGeeks' d=2  Lfirst = input[0 : d]  Lsecond = input[d :]  Rfirst = input[0 : len(input)-d] Rsecond = input[len(input)-d : ] print ("Left Rotation : ", (Lsecond + Lfirst) ) print ("Right Rotation : ", (Rsecond + Rfirst)) |

1. Find all duplicate characters in string

|  |
| --- |
| string = input() dups=[] for ch in string:  if string.count(ch)>1 and ch not in dups:  dups.append(ch)  print('The duplicate characters are {}'.format(dups)) |

ASSIGNMENT

Q-1. What Will Be The Output Of The Following Code Snippet?

a = {(1,2):1,(2,3):2} print(a[1,2])

## ANS- A. KEY ERROR

-2. What Will Be The Output Of The Following Code Snippet? a = {'a':1,'b':2,'c':3} print (a['a','b']) print(a.get(‘a’,’b’))

## ANS- A. KEY ERROR

Q-3. What Will Be The Output Of The Following Code Snippet? fruit = {} def addone(index): if index in fruit: fruit[index] += 1 else: fruit[index] = 1 addone('Apple') addone('Banana') addone('apple') print (len(fruit))

ANS- C. 3

Q-4. What Will Be The Output Of The Following Code Snippet? arr = {} arr[1] = 1 arr['1'] = 2 arr[1] += 1 sum = 0

for k in arr: sum += arr[k] print (sum)

ANS- D. 4

Q-5. What Will Be The Output Of The Following Code Snippet?

my\_dict = {} my\_dict[1] = 1 my\_dict['1'] = 2 my\_dict[1.0] = 4 sum = 0 for k in my\_dict: sum += my\_dict[k] print (sum)

ANS- A. 7

Q-6. What Will Be The Output Of The Following Code Snippet?

my\_dict = {} my\_dict[(1,2,4)] = 8 my\_dict[(4,2,1)] = 10 my\_dict[(1,2)] = 12 sum = 0 for k in my\_dict: sum += my\_dict[k] print (sum) print(my\_dict)

## ANS- B. 30 {(1, 2): 12, (4, 2, 1): 10, (1, 2, 4): 8}

Q-7. What Will Be The Output Of The Following Code Snippet?

box = {} jars = {} crates = {} box['biscuit'] = 1 box['cake'] = 3 jars['jam'] = 4 crates['box'] = box crates['jars'] = jars

print (len(crates[box]))

ANS- D. TYPE ERROR

Q-8. What Will Be The Output Of The Following Code Snippet? dict = {'c': 97, 'a': 96, 'b': 98} for \_ in sorted(dict):

print (dict[\_])

## ANS- A. 96 98 97

Q-9. What Will Be The Output Of The Following Code Snippet? rec = {"Name" : "Python", "Age":"20"}

r = rec.copy()

print(id(r) == id(rec))

## ANS- B. FALSE

Q-10. What Will Be The Output Of The Following Code Snippet?

rec = {"Name" : "Python", "Age":"20", "Addr" : "NJ", "Country" : "USA"} id1 = id(rec) del rec

rec = {"Name" : "Python", "Age":"20", "Addr" : "NJ", "Country" : "USA"} id2 = id(rec)

print(id1 == id2)

ANS- B. FALSE

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

# ANS-

my\_dict = {"apple": 5, "banana": 10, "orange": 3, "pear": 8} asc\_dict = dict(sorted(my\_dict.items(), key=lambda x: x[1]))

desc\_dict = dict(sorted(my\_dict.items(), key=lambda x: x[1], reverse=True)) print("Ascending order:", asc\_dict)

print("Descending order:", desc\_dict)

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}

## ANS-

sample\_dict = {0: 10, 1: 20} key = 2 value = 30

sample\_dict[key] = value

print(sample\_dict)

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}

# ANS-

dic1 = {1: 10, 2: 20} dic2 = {3: 30, 4: 40} dic3 = {5: 50, 6: 60} result = {} for d in (dic1, dic2, dic3): result.update(d)

print(result)

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

## ANS-

my\_dict = {'apple': 2, 'banana': 4, 'orange': 6} key = 'banana' if key in my\_dict:

print(f"The key '{key}' exists in the dictionary.") else: print(f"The key '{key}' does not exist in the dictionary.")

5. Write a Python program to iterate over dictionaries using for loops.

## ANS-

my\_dict = {"name": "John", "age": 30, "city": "New York"} for key in my\_dict: print(key, ":", my\_dict[key])

6. 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}

ANS-

n = 5 d = {} for x in range(1, n+1): d[x] = x\*x

print(d)

7. 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}

## ANS-

d = {} for i in range(1, 16):

d[i] = i \*\* 2

print(d)

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

## ANS-

dict1 = {'a': 1, 'b': 2} dict2 = {'c': 3, 'd': 4}

dict1.update(dict2)

print(dict1) # {'a': 1, 'b': 2, 'c': 3, 'd': 4}

9. Write a Python program to iterate over dictionaries using for loops.

## ANS-

dict = {'name': 'John', 'age': 25, 'country': 'USA'} for key in dict: print(key, ':', dict[key])

10. Write a Python program to sum all the items in a dictionary.

## ANS-

my\_dict = {'a': 10, 'b': 20, 'c': 30} total = sum(my\_dict.values())

print(total)

11. Write a Python program to multiply all the items in a dictionary

## ANS-

def multiply\_dict\_items(dict):

result = 1 for key in dict: result \*= dict[key] return result

12. Write a Python program to remove a key from a dictionary.

# ANS-

my\_dict = {'apple': 2, 'banana': 3, 'orange': 4}

print("Original dictionary:", my\_dict)

13. Write a Python program to map two lists into a dictionary.

## ANS-

keys = ['name', 'age', 'gender'] values = ['Alice', 25, 'Female'] my\_dict = dict(zip(keys, values)) print(my\_dict)

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

## ANS-

my\_dict = {'b': 2, 'c': 3, 'a': 1}

sorted\_dict = {k: v for k, v in sorted(my\_dict.items(), key=lambda x: x[0])} print(sorted\_dict)

15. Write a Python program to get the maximum and minimum value in a dictionary.

## ANS-

my\_dict = {'a': 10, 'b': 5, 'c': 20, 'd': 30} max\_val = max(my\_dict.values()) min\_val = min(my\_dict.values()) print('Maximum value:', max\_val)

print('Minimum value:', min\_val)

16. Write a Python program to get a dictionary from an object's fields.

## ANS-

class Person: def \_\_init\_\_(self, name, age): self.name = name

self.age = age

person = Person("John", 30)

person\_dict = vars(person)

print(person\_dict)

17. Write a Python program to remove duplicates from Dictionary

## ANS-

original\_dict = {"a": 1, "b": 2, "c": 1, "d": 3, "e": 2}

new\_dict = {value:key for key, value in original\_dict.items()}

18. Write a Python program to check a dictionary is empty or not.

## ANS-

my\_dict = {}

if len(my\_dict) == 0:

print("The dictionary is empty") else:

print("The dictionary is not empty")

19. 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})

## ANS-

from collections import Counter

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

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

result = Counter(d1) + Counter(d2)

print(result)

20. Write a Python program to print all unique values in a dictionary. Sample Data : [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"},

{"VI": "S005"}, {"VII":"S005"},

{"V":"S009"},{"VIII":"S007"}]

Expected Output : Unique Values: {'S005',

'S002', 'S007', 'S001', 'S009'}

## ANS-

data = [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S005"}, {"V":"S009"},{"VIII":"S007"}]

unique\_values = set() for d in data: for value in d.values():

unique\_values.add(value)

print("Unique Values:", unique\_values)

21. Write a Python program to create and display all combinations of letters, selecting each letter from a different key in a dictionary.

Sample data : {'1':['a','b'], '2':['c','d']}

Expected Output:

ac ad bc

bd

## ANS-

def generate\_combinations(data):

combinations = [] keys = data.keys() values = [data[key] for key in keys] for i in range(len(values[0])): for j in range(len(values[1])): combination = values[0][i] + values[1][j] combinations.append(combination) return combinations

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

## ANS-

def find\_highest\_values(data):

sorted\_values = sorted(data.values(), reverse=True) return sorted\_values[:3]

23. Write a Python program to combine values in python list of dictionaries.

Sample data: [{'item': 'item1', 'amount':

400}, {'item': 'item2', 'amount': 300}, {'item':

'item1', 'amount': 750}]

Expected Output: Counter({'item1': 1150,

'item2': 300})

## ANS-

from collections import Counter def combine\_values(data):

result = Counter() for d in data: result[d['item']] += d['amount'] return result

24. Write a Python program to create a dictionary from a string.

Note: Track the count of the letters from the string.

Sample string : 'w3resource'

Expected output: {'3': 1, 's': 1, 'r': 2, 'u': 1,

'w': 1, 'c': 1, 'e': 2, 'o': 1}

## ANS-

def create\_dict\_from\_string(s): counts = {} for letter in s: if letter in counts:

counts[letter] += 1 else:

counts[letter] = 1

return counts

25. Write a Python program to print a dictionary in table format.

## ANS-

def print\_dict\_table(data): keys = list(data.keys()) values = list(data.values()) max\_key\_length = max(len(str(key)) for key in keys) max\_value\_length = max(len(str(value)) for value in values) print('+' + '-' \* (max\_key\_length + 2) + '+' + '-' \* (max\_value\_length + 2) + '+') print('| {:<{}} | {:>{}} |'.format('Key', max\_key\_length, 'Value', max\_value\_length)) print('+' + '-' \* (max\_key\_length + 2) + '+' + '-' \* (max\_value\_length + 2) + '+') for key, value in data.items(): print('| {:<{}} | {:>{}} |'.format(key, max\_key\_length, value, max\_value\_length)) print('+' + '-' \* (max\_key\_length + 2) + '+' + '-' \* (max\_value\_length + 2) + '+')

26. Write a Python program to count the values associated with key in a dictionary. Sample data: = [{'id': 1, 'success': True, 'name': 'Lary'}, {'id': 2, 'success': False, 'name':

'Rabi'}, {'id': 3, 'success': True, 'name':

'Alex'}]

Expected result: Count of how many

dictionaries have success as True

## ANS-

def count\_dicts\_with\_value(data, key, value):

count = 0 for dictionary in data: if dictionary.get(key) == value:

count += 1

return count

27. Write a Python program to convert a list into a nested dictionary of keys.

## ANS-

def list\_to\_nested\_dict(lst):

nested\_dict = {} for item in reversed(lst): nested\_dict = {item: nested\_dict} return nested\_dict

28. Write a Python program to sort a list alphabetically in a dictionary

## ANS-

my\_dict = {'fruit': ['apple', 'orange', 'banana', 'kiwi']} my\_dict['fruit'].sort()

print(my\_dict)

29. Write a Python program to remove spaces from dictionary keys.

## ANS-

my\_dict = {'my key': 1, 'another key': 2, 'third key': 3}

new\_dict = {key.replace(' ', ''): value for key, value in my\_dict.items()} print('Original dictionary:', my\_dict)

print('New dictionary:', new\_dict)

30. 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 ANS-

shop\_inventory = {'item1': 45.50, 'item2': 35, 'item3': 41.30, 'item4': 55, 'item5': 24} top\_three = sorted(shop\_inventory.items(), key=lambda x: x[1], reverse=True)[:3] for item, price in top\_three:

print(item, price)

1. Write a Python program to get the key, value and item in a dictionary.

ANS-

my\_dict = {'apple': 2, 'banana': 3, 'orange': 4} for key, value in my\_dict.items():

item = (key, value) print(f"Key: {key}, Value: {value}, Item: {item}")

1. Write a Python program to print a dictionary line by line.

ANS-

my\_dict = {'apple': 2, 'banana': 3, 'orange': 4} for key, value in my\_dict.items():

print(f"{key}: {value}")

1. Write a Python program to check multiple keys exists in a dictionary.

ANS-

my\_dict = {'apple': 2, 'banana': 3, 'orange': 4, 'pear': 5} keys\_to\_check = ['apple', 'banana', 'peach'] if all(key in my\_dict for key in keys\_to\_check):

print("All keys exist in the dictionary") else:

print("At least one key does not exist in the dictionary")

1. Write a Python program to count number of items in a dictionary value that is a list.

ANS-

my\_dict = {'fruits': ['apple', 'banana', 'orange'], 'vegetables': ['carrot', 'celery']} for key, value in my\_dict.items():

num\_items = len(value) print(f"{key} has {num\_items} items")

1. Write a Python program to sort Counter by value.

Sample data : {'Math':81, 'Physics':83,

'Chemistry':87}

Expected data: [('Chemistry', 87),

('Physics', 83), ('Math', 81)]

ANS-

from collections import Counter

my\_counter = Counter({'Math':81, 'Physics':83, 'Chemistry':87})

sorted\_counter = sorted(my\_counter.items(), key=lambda x: x[1], reverse=True) print(sorted\_counter)

1. Write a Python program to create a dictionary from two lists without losing duplicate values.

Sample lists: ['Class-V', 'Class-VI', 'Class-

VII', 'Class-VIII'], [1, 2, 2, 3]

Expected Output: defaultdict(<class 'set'>, {'Class-VII': {2}, 'Class-VI': {2}, 'Class-VIII':

{3}, 'Class-V': {1}})

ANS-

from collections import defaultdict keys = ['Class-V', 'Class-VI', 'Class-VII', 'Class-VIII'] values = [1, 2, 2, 3] result = defaultdict(set) for k, v in zip(keys, values):

result[k].add(v) print(result)

1. Write a Python program to replace dictionary values with their sum.

ANS-

data = {'a': 10, 'b': 20, 'c': 30, 'd': 40} total = sum(data.values()) for key in data: data[key] = total print(data)

1. Write a Python program to match key values in two dictionaries.

### Sample dictionary: {'key1': 1, 'key2': 3,

'key3': 2}, {'key1': 1, 'key2': 2}

Expected output: key1: 1 is present in both x and y

ANS-

x = {'key1': 1, 'key2': 3, 'key3': 2} y = {'key1': 1, 'key2': 2} for key in x: if key in y: if x[key] == y[key]:

print(key + ": " + str(x[key]) + " is present in both x and y")

Q1- Write a program to guess the correct number.

Ans:

import random

# generate a random number between 1 and 100 secret\_number = random.randint(1, 100)

# initialize the number of guesses num\_guesses = 0

while True:

# get a guess from the user guess = int(input("Guess a number between 1 and 100: "))

# increment the number of guesses num\_guesses += 1 # check if the guess is correct if guess == secret\_number: print(f"Congratulations, you guessed the number in

{num\_guesses} guesses!") break

# give the user a hint

if guess < secret\_number:

print("Too low!")

else: print("Too high!")

Q2-Write a program for rock,paper,scissor (computer vs human) Ans:

import random print("Welcome to Rock, Paper, Scissors!")

# define the options options = ["rock", "paper", "scissors"] # loop until the user decides to quit while True:

# get the user's choice user\_choice = input("Enter rock, paper, scissors, or q to quit:

").lower()

# check if the user wants to quit if user\_choice == "q": print("Thanks for playing!") break

# check if the user's choice is valid if user\_choice not in options:

print("Invalid choice, please try again.") continue

# generate the computer's choice computer\_choice = random.choice(options)

# print the choices print("You chose:", user\_choice) print("The computer chose:", computer\_choice)

# determine the winner if user\_choice == computer\_choice: print("It's a tie!")

elif (user\_choice == "rock" and computer\_choice ==

"scissors") or \

(user\_choice == "paper" and computer\_choice == "rock")

or\

(user\_choice == "scissors" and computer\_choice == "paper"): print("You win!")

else: print("The computer wins!")

Q3- Write a program to generate password with a fixed length.

Ans:

import random import string def generate\_password(length):

"""Generate a random password with the given length""" # define the set of characters to choose from chars = string.ascii\_letters + string.digits +

string.punctuation

# generate the password password = "" for i in range(length):

password += random.choice(chars) return password

# get the desired length from the user

length = int(input("Enter the length of the password: "))

# generate the password and print it password = generate\_password(length)

print("Your password is:", password)

Q4- Write a program to roll the dice till the 6 number is not appear. (computer vs human) Ans:

import random print("Welcome to the Dice Rolling Game!")

# loop until a 6 is rolled while True:

# prompt the user to roll the dice input("Press enter to roll the dice...")

# roll the dice for the user and computer user\_roll = random.randint(1, 6) computer\_roll = random.randint(1, 6)

# print the rolls print("You rolled:", user\_roll) print("The computer rolled:", computer\_roll)

# check if a 6 was rolled if user\_roll == 6 or computer\_roll == 6:

break # determine the winner if user\_roll == 6 and computer\_roll != 6:

print("Congratulations, you win!")

elif user\_roll != 6 and computer\_roll == 6:

print("Sorry, the computer wins!")

else:

print("It's a tie!")

1. **Write a Python program to read an entire text file.**

def read\_text\_file(file\_path):

try:

with open(file\_path, 'r') as file:

contents = file.read()

return contents

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

file\_contents = read\_text\_file(file\_path)

if file\_contents:

print(file\_contents)

**2. Write a Python program to read first n lines of a file**

def read\_first\_n\_lines(file\_path, n):

try:

with open(file\_path, 'r') as file:

lines = file.readlines()

first\_n\_lines = "".join(lines[:n])

return first\_n\_lines

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

n = 5 # Specify the number of lines to read

first\_n\_lines = read\_first\_n\_lines(file\_path, n)

if first\_n\_lines:

print(first\_n\_lines)

1. **Write a Python program to append text to a file and display the text.**

def append\_text\_to\_file(file\_path, text):

try:

with open(file\_path, 'a') as file:

file.write(text)

print("Text appended successfully.")

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while writing to the file '{file\_path}'.")

def read\_file\_contents(file\_path):

try:

with open(file\_path, 'r') as file:

contents = file.read()

return contents

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

text\_to\_append = "This is the new text to append."

# Append text to the file

append\_text\_to\_file(file\_path, text\_to\_append)

# Read and display the updated contents of the file

file\_contents = read\_file\_contents(file\_path)

if file\_contents:

print(file\_contents)

1. **Write a Python program to append text to a file and display the text.**

def append\_text\_to\_file(file\_path, text):

try:

with open(file\_path, 'a') as file:

file.write(text)

print("Text appended successfully.")

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while writing to the file '{file\_path}'.")

def read\_file\_contents(file\_path):

try:

with open(file\_path, 'r') as file:

contents = file.read()

return contents

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

text\_to\_append = "This is the new text to append."

# Append text to the file

append\_text\_to\_file(file\_path, text\_to\_append)

# Read and display the updated contents of the file

file\_contents = read\_file\_contents(file\_path)

if file\_contents:

print(file\_contents)

1. **Write a Python program to read last n lines of a file.**

def read\_last\_n\_lines(file\_path, n):

try:

with open(file\_path, 'r') as file:

lines = file.readlines()

last\_n\_lines = "".join(lines[-n:])

return last\_n\_lines

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

n = 5 # Specify the number of lines to read from the end

last\_n\_lines = read\_last\_n\_lines(file\_path, n)

if last\_n\_lines:

print(last\_n\_lines)

1. **Write a Python program to read a file line by line and store it into a list**

def read\_file\_lines(file\_path):

try:

with open(file\_path, 'r') as file:

lines = file.readlines()

lines = [line.strip() for line in lines]

return lines

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

lines\_list = read\_file\_lines(file\_path)

if lines\_list:

for line in lines\_list:

print(line)

**6. Write a Python program to read a file line by line store it into a variable**

def read\_file\_contents(file\_path):

try:

with open(file\_path, 'r') as file:

contents = file.read()

return contents

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

file\_contents = read\_file\_contents(file\_path)

if file\_contents:

print(file\_contents)

1. **Write a Python program to read a file line by line store it into an array**

def read\_file\_lines(file\_path):

try:

with open(file\_path, 'r') as file:

lines\_array = []

for line in file:

lines\_array.append(line.strip())

return lines\_array

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

lines\_array = read\_file\_lines(file\_path)

if lines\_array:

for line in lines\_array:

print(line)

1. **Write a python program to find the longest words**

def find\_longest\_words(file\_path):

try:

with open(file\_path, 'r') as file:

words = file.read().split()

longest\_words = []

max\_length = 0

for word in words:

word\_length = len(word)

if word\_length > max\_length:

max\_length = word\_length

longest\_words = [word]

elif word\_length == max\_length:

longest\_words.append(word)

return longest\_words

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

longest\_words = find\_longest\_words(file\_path)

if longest\_words:

print("Longest word(s):")

for word in longest\_words:

print(word)

1. **Write a Python program to count the number of lines in a text file.**

def count\_lines(file\_path):

try:

with open(file\_path, 'r') as file:

line\_count = sum(1 for line in file)

return line\_count

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

line\_count = count\_lines(file\_path)

if line\_count is not None:

print(f"Number of lines in the file: {line\_count}")

1. Write a Python program to count the frequency of words in a file

from collections import Counter

def count\_word\_frequency(file\_path):

try:

with open(file\_path, 'r') as file:

words = file.read().split()

word\_frequency = Counter(words)

return word\_frequency

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while reading the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/text/file.txt'

word\_frequency = count\_word\_frequency(file\_path)

if word\_frequency:

for word, frequency in word\_frequency.items():

print(f"Word: {word}, Frequency: {frequency}")

1. **Write a Python program to get the file size of a plain file**

import os

def get\_file\_size(file\_path):

try:

file\_size = os.path.getsize(file\_path)

return file\_size

except FileNotFoundError:

print(f"File '{file\_path}' not found.")

except IOError:

print(f"An error occurred while accessing the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/file.txt'

size = get\_file\_size(file\_path)

if size is not None:

print(f"File size: {size} bytes")

1. **Write a Python program to write a list to a file**

def write\_list\_to\_file(file\_path, data\_list):

try:

with open(file\_path, 'w') as file:

for item in data\_list:

file.write(str(item) + '\n')

print("List written to file successfully.")

except IOError:

print(f"An error occurred while writing to the file '{file\_path}'.")

# Example usage

file\_path = 'path/to/your/output/file.txt'

data\_list = [1, 2, 3, 4, 5]

write\_list\_to\_file(file\_path, data\_list)

1. **Write a Python program to copy the contents of a file to another file**

def copy\_file(source\_file\_path, destination\_file\_path):

try:

with open(source\_file\_path, 'r') as source\_file:

with open(destination\_file\_path, 'w') as destination\_file:

for line in source\_file:

destination\_file.write(line)

print("File copied successfully.")

except FileNotFoundError:

print(f"One or both of the files '{source\_file\_path}', '{destination\_file\_path}' not found.")

except IOError:

print(f"An error occurred while copying the files.")

# Example usage

source\_file\_path = 'path/to/your/source/file.txt'

destination\_file\_path = 'path/to/your/destination/file.txt'

copy\_file(source\_file\_path, destination\_file\_path)

**14. Write a Python program to combine each line from first file with the**

**corresponding line in second file**

def combine\_files(file1\_path, file2\_path, output\_file\_path):

try:

with open(file1\_path, 'r') as file1, open(file2\_path, 'r') as file2, open(output\_file\_path, 'w') as output\_file:

for line1, line2 in zip(file1, file2):

combined\_line = line1.strip() + " " + line2.strip()

output\_file.write(combined\_line + "\n")

print("Files combined successfully.")

except FileNotFoundError:

print(f"One or more of the files '{file1\_path}', '{file2\_path}', '{output\_file\_path}' not found.")

except IOError:

print("An error occurred while combining the files.")

# Example usage

file1\_path = 'path/to/your/file1.txt'

file2\_path = 'path/to/your/file2.txt'

output\_file\_path = 'path/to/your/output.txt'

combine\_files(file1\_path, file2\_path, output\_file\_path)