1. Write a program to reverse a string.

Sample input: “1234abcd”

Expected output: “dcba4321”

string="1234abcd"  
print(string[::-1])

2. Write a function that accepts a string and prints the number of uppercase letters and lowercase

letters.

Sample input: “abcSdefPghijQkl”

Expected Output: No. of Uppercase characters : 3 No. of Lower case Characters : 12

def string\_test(sample1):  
 d = {"UPPER\_CASE": 0, "LOWER\_CASE": 0}  
 for i in sample1:  
 if i.isupper():  
 d["UPPER\_CASE"] += 1  
 elif i.islower():  
 d["LOWER\_CASE"] += 1  
 else :  
 pass  
 print("original\_String:",sample1)  
 print("UPPER\_CASE:", d["UPPER\_CASE"])  
 print("LOWER\_CASE:", d["LOWER\_CASE"])  
string\_test("abcSdefPghijQkl")

3. Create a function that takes a list and returns a new list with unique elements of the first list.

def unique\_list(l):  
 x=[]  
 for i in l:  
 if i not in x:  
 x.append(i)  
 return(x)  
print(unique\_list([1,2,2,3,3,4,5]))

4. Write a program that accepts a hyphen-separated sequence of words as input and prints the words

in a hyphen-separated sequence after sorting them alphabetically.

item=[n for n in input().split("-")]  
item.sort()  
  
print("-".join(item))

5. Write a program that accepts a sequence of lines as input and prints the lines after making all

characters in the sentence capitalized.

Sample input: Hello world Practice makes man perfect

Expected output: HELLO WORLD PRACTICE MAKES MAN PERFECT

s=input("enter a line:")  
d=s.upper()  
print(d)

6. Define a function that can receive two integral numbers in string form and compute their sum and

print it in the console.

def calculateSum(a,b):  
 s=int(a)+int(b)  
 return s  
num1="10"  
num2="2"  
sum=calculateSum(num1,num2)  
print("sum:",sum)

7. Define a function which can generate and print a tuple where the values are square of numbers

between 1 and 20 (both 1 and 20 included).

def printValue():  
 l=list()  
 for i in range(1,21):  
 l.append(i\*\*2)  
 print(l)  
printValue()

9. Write a function called showNumbers that takes a parameter called limit. It should print all the

numbers between 0 and limit with a label to identify the even and odd numbers.

Sample input: show Numbers(3) (where limit=3)

Expected output:

0 EVEN

1 ODD

2 EVEN

3 ODD

def showNumbers(l):  
 for i in range(l+1):  
 if i%2==0:  
 print(i,"even")  
 elif i%2!=0:  
 print(i,"odd")  
showNumbers(4)

10. Write a program which uses filter() to make a list whose elements are even numbers between 1

and 20 (both included)

|  |
| --- |
|  |
| even\_number = list(filter(lambda x:x %2 ==0,[i for i in range(1,21)]))  print(even\_number)  11. Write a program which uses map() and filter() to make a list whose elements are squares of even  numbers in [1,2,3,4,5,6,7,8,9,10].  Hints: Use filter() to filter even elements of the given listUse map() to generate a list of squares of the  numbers in the filtered list. Use lambda() to define anonymous functions.  list1 = [1,2,3,4,5,6,7,8,9,10]  even\_num = list(filter(lambda i :i%2 ==0,list1)) square\_num = list(map(lambda j: j\*j, even\_num)) print(even\_num) print(square\_num) |  |

12. Write a function to compute 5/0 and use try/except to catch the exceptions

=int(input("value a:"))  
b=int(input("value b:"))  
try:  
 div=a/b  
 print(div)  
except exception:  
 print("we cannot divide by zero")  
 print("hello")

13. Flatten the list [1,2,3,4,5,6,7] into 1234567 using reduce().

14. 14. Write a program in Python to find the values which are not divisible by 3 but are a multiple of 7.

Make sure to use only higher order functions.

s= list(filter(lambda x: x%3!=0 and x%7==0,range(100)))  
print(s)

15.Write a program in Python to multiply the elements of a list by itself using a traditional function

and pass the function to map() to complete the operation.

def multiply(n):  
 return n\*n  
  
numbers=(1,2,3,4)  
result=map(multiply,numbers)  
print(list(result))

16. What is the output of the following codes:

(i) def foo():

try:

return 1

finally:

return 2

k = foo()

print(k)

IndentationError: expected an indented block

(ii)def a():

try:

f(x, 4)

finally:

print('after f')

print('after f?')

a()

cannot identified f element