PYTHON ASSIGNMENT

MAKE A MOVE TO PYTHON

ASSIGNMENTS

SUBMITTED TO

YASHIKA KHATRI

By: ANKUR SINGH

TASK-4:
FUNCTIONS

Write a program to reverse a string.

Sample data: “1234abcd”

Expected Output: “dcba4321”

```
def reverse(string):
   return string[::-1]

print(reverse("1234abcd"))
```

Output:

dcba4321

Write a function that accepts a string and calculate the number of uppercase letters and lowercase letters.
Expected Output:
No. of Upper case characters : 3
No. of Lower case Characters : 12

```
def case(string):
   d={"UpperCase":0, "LowerCase":0}
   for data in string:
      if data.isupper():
         d["UpperCase"] = d["UpperCase"] + 1
      elif data.islower():
         d["LowerCase"] = d["LowerCase"] + 1
      else:
         pass

   print ("Total Upper case characters are : {0} ".format(d["UpperCase"]))
   print ("Total Lower case Characters : {0}".format(d["LowerCase"]))

case('ConsultADD is the Best')
```

Output:

Total Upper case characters are : 5
Total Lower case Characters : 14

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

```
def list(l):
   new_list = []
   for data in l:
      if data not in new_list:
         new_list.append(data)
   return new_list

print(list([1,2,3,4,5,6,4,6,2,1,4,5]))
```

Output:

[1, 2, 3, 4, 5, 6]

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.

```
def convert(user_input):
   user_input = user_input.split("-")
   user_input = sorted(user_input)
   data = "-".join(user_input)
   return data

print(convert(user_input = input("Enter the words seperated by hyphens:
")))
```

Output:

Enter the words seperated by hyphens: Consultadd-is-the-best-training-program
Consultadd-best-is-program-the-training

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 perfect
Expected Output:
HELLO WORLD
PRACTICE MAKES PERFECT

```
def capital():
  list = []
  while True:
    user_input = input()
    if user_input:
      user_input = user_input.upper()
      list.append(user_input)
    else:
      break

  data = "\n".join(list)
  return data

print(capital())
```

Output:

hello world
practice makes perfect

HELLO WORLD
PRACTICE MAKES PERFECT

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

```
def compute(num1, num2):
   sum = int(num1) + int(num2)

   return sum
print(compute("5", "6"))
```

Output:

11

Define a function that can accept two strings as input and print the string with maximum length in console. If two strings have the same length, then the function should print all strings line by line.

```
def maximum(string1,string2):

   len1 = len(string1)
   len2 = len(string2)

   if len1 > len2:
      return ("{0}".format(string1))
   elif len2 > len1:
      return ("{0}".format(string2))
   else:
      return (string1+'\n'+string2)



string1 = input("Enter 1st String: ")
string2 = input("Enter 2nd String: ")

print(maximum(string1, string2))
```

Output:

Enter 1st String: one
Enter 2nd String: ninety
Ninety

Define a function which can generate and print a tuple where the value are square of numbers between 1 and 20.

```
def square():
   original_list = []
   squared_list = []

   for data in range(1, 21):
      original_list.append(data)
      squared_list.append(data ** 2)


   print(tuple(original_list))
   print(tuple(squared_list))

square()
```

Output:

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20)
(1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400)

Write a function called show Numbers 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.

```
def showNumbers(limit):
    for data in range(0, limit + 1):
        if data % 2 == 0:
            print("{0}".format(data) + " EVEN")
        else:
            print("{0}".format(data) + " ODD")


showNumbers(limit=int(input("Please enter the limit: ")))
```

Output:

```
0 EVEN
1 ODD
2 EVEN
3 ODD
4 EVEN
```

Write a program which can filter() to make a list whose elements are even number between 1 and 20 (both included)

```
def even(data):
    return data % 2 == 0

number = filter(even, range(1,21))

print(list(number))
```

Output:

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

Write a program which can map() and filter() to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10]
Hints: Use map() to generate a list.
Use filter() to filter elements of a list
Use lambda to define anonymous functions

```
def even(data):
   return data % 2 ==0


list_new = [1,2,3,4,5,6,7,8,9,10]

list_new = map(lambda data: data ** 2,filter(even,list_new))

print(list(list_new))
```

Output:

```
[4, 16, 36, 64, 100]
```

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

```
def division():
   try:
      a = 5
      b = 0

      c = a/b

   except (Exception) as e:
      print("Dividing a number by zero is not allowed! :%s"%e)

division()
```

Output:

Dividing a number by zero is not allowed! :division by zero

Flatten the list [[1,2,3],[4,5],[6,7,8]] into [1,2,3,4,5,6,7,8] using reduce Goal : Turn [1,2,3,4,5,6,7] to 1234567

```
from functools import reduce

def add(a, b):
   return a + b

lists = [[1, 2, 3], [4, 5], [6, 7, 8]]
new_list = reduce(add, lists)

print(new_list)
```

Output:

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

What is the output of the following codes:

(i)

```
def foo():
    try:
        return 1
    finally:
        return 2
k = foo()
print(k)
```

Output: 2
Reason: Here the finally block is executed. This is because finally will be executed every time even if try block is there.

(ii)
```
def a():
    try:
        f(x, 4)
    finally:
        print('after f')
    print('after f?')
a()
```

Output: NameError: name 'f' is not defined.

Reason: Here we have got “NameError” because “f” is not defined. Here we were using try and finally to handle exceptions.