P E S University

Department of Computer Science & Engineering

Session : Aug-Dec 2019

Introduction to Computing using Python Laboratory(UE19CS102)

Week 11– Programs on Functions

1

Write a function that accepts a string and calculate the number of upper case letters and lower case letters.

Solution:

```
def string_test(s):
    d={"UPPER_CASE":0, "LOWER_CASE":0}
    for c in s:
        if c.isupper():
            d["UPPER_CASE"]+=1
        elif c.islower():
            d["LOWER_CASE"]+=1
        else:
            pass
    print ("Original String : ", s)
    print ("No. of Upper case characters : ", d["UPPER_CASE"])
    print ("No. of Lower case Characters : ", d["LOWER_CASE"])

string_test('This is python lab program')
```

2

```
def is_square(x) :
        pass
```

check whether a given number is a perfect square.

```
def is_even(x) :
        pass
```

check whether a given number is an even number.

```
# this could be another set of functions!
```

find all numbers between 1 and n which are
both square and even.

Solution:

```
def is_square(x) :
          i = 1
          while i * i < x :
```

```
                i += 1
        return i * i == x

def is_even(x) :
        return x % 2 == 0

n = 25
print(is_square(n), is_even(n)) # True False
n = 16
print(is_square(n), is_even(n)) # True True
n = 15
print(is_square(n), is_even(n)) # False False
n = 20
print(is_square(n), is_even(n)) # False True
for i in range(1, n + 1):
        if is_even(i) and is_square(i) :
                print("square and even : ", i)
```

3

Solve the following using Recursion:
a) find the length of a string
b) find the smallest element in a list
c) reverse a string
d) to compute a power b.

Solution:

a) find the length of a string.

```
def length(s):
        if s == '':
                return 0
        else:
                return 1 + length(s[:-1])
s = input("Enter a string: ")
print("Length of a given string ", s, " is ",length(s))
```

b) find the smallest element in a list.

```
def Min(list):
        if len(list) <= 1:
                return list[0]
        else:
                m = Min(list[1:])
                return m if m < list[0] else list[0]
l = [69,12,54,38,71,2]
print("Minimum element in a list ", l, " is ",Min(l))
```

c) reverse a string.

```
def reverse(s):
        if s == "":
                return s
        else:
                return reverse(s[1:]) + s[0]
s = input("Enter a string: ")
print("Reverse of a string ", s, " is ",reverse(s))
```

d) to compute a to th power b.

```
def power(a,b):
    if b==0:
        return 1
    elif a==0:
        return 0
    elif b==1:
        return a
    else:
        return a*power(a,b-1)
print(power(3,4))
```

4 Information of a student - srn, name, marks (highest 100) in P, C, M are stored in a tuple.
Information of a group of students is stored in a list of tuple.
a) sort the list and display
b) sort the list based on the name and display
c) sort the list based on the total of P C M marks in descending order and display

Soluton:

a) sort the list and display.

```
s = [("890","x",(95,78,99)),("123","a",(90,98,89)),("567","p",(59,68,100))]
srn = sorted(s)
print("Student list is sorted based on 1st field, SRN:\n", srn)\
```

b) sort the list based on the name and display.

```
s = [("890","x",(95,78,99)),("123","a",(90,98,89)),("567","p",(59,68,100))]
name = sorted(s, key = lambda t: t[1])
print("Student list is sorted based on 2nd field, Name:\n", name)
```

c) sort the list based on the total of P C M marks in descending order and display.

```
s = [("890","x",(95,78,99)),("123","a",(90,98,89)),("567","p",(59,68,100))]
PCM = sorted(s, reverse = True, key = lambda t: sum(t[2]))
print("Student list is sorted based on total of PCM marks in descending order:\n", PCM)
```

5 Given a list of strings,

a) find the longest string.

b) find all strings ending with a given suffix.

Solutions:

a) find the longest string.

```
string = input("Enter a string : ").split()
print(string)
long = max(string, key = lambda x : len(x))
print("The longest string is ",long)
```

b) find all strings ending with a given suffix.

```
string = input("Enter a string : ").split(" ")
print(string)
suffix = input("Enter a suffix: ")
l = list(filter(lambda string:string.endswith(suffix), string))
print("Strings ending with suffix", suffix, " are ", l)
```