University of Cape Town ~~ Department of Computer Science

Computer Science 1015F ~~ 2016

Class Test 3

Enter the following details AND shade in the corresponding blocks to the right with your Student Number.

Faculty (please tick one):

Science	Engineering	Commerce	Humanities	Other:

Student Number :

____

Name (optional) :

____

Marks : 40

Time : 40 minutes

Instructions:

a) Answer all questions.

b) Write your answers in PEN in the spaces provided.

c) Show all calculations where applicable.

A	□	□	□	□	□	□	□	□	□	0
B	□	□	□	□	□	□	□	□	□	1
C	□	□	□	□	□	□	□	□	□	2
D	□	□	□	□	□	□	□	□	□	3
E	□	□	□	□	□	□	□	□	□	4
F	□	□	□	□	□	□	□	□	□	5
G	□	□	□	□	□	□	□	□	□	6
H	□	□	□	□	□	□	□	□	□	7
I	□	□	□	□	□	□	□	□	□	8
J	□	□	□	□	□	□	□	□	□	9
K	□	□	□	□	□	□				
L	□	□	□	□	□	□				
M	□	□	□	□	□	□				
N	□	□	□	□	□	□				
O	□	□	□	□	□	□				
P	□	□	□	□	□	□				
Q	□	□	□	□	□	□				
R	□	□	□	□	□	□				
S	□	□	□	□	□	□				
T	□	□	□	□	□	□				
U	□	□	□	□	□	□				
V	□	□	□	□	□	□				
W	□	□	□	□	□	□				
X	□	□	□	□	□	□				
Y	□	□	□	□	□	□				
Z	□	□	□	□	□	□				

FOR OFFICIAL USE ONLY:

Question		1		2		3		4		5		6		7		8	
Max		23		17													
Marks	0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
	1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
	2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
	3	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
	4	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
	5	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
	6	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
	7	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
	8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
	9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Marker																	

Question 1 – Arrays, Dictionaries and Files [23]

Examine the `Q1.py` module listed at the end of the test and answer the following questions.

(i) Explain what happens in the `xWords(wordList)` function if:

A. the file "`input.txt`" does not exist in the current directory. [1]

____

B. the file "out`put.txt`" does not exist in the current directory. [1]

____

(ii) Describe briefly, and in clear English, what the function `xWords(wordList)` does and what output is produced. Your answer must consider parameters of different types. [5]

____

____

____

____

____

____

____

____

____

____

____

____

____

____

(iii) Write down the exact output when `Q1.py` is run in the in the Python3 interpreter. [6]

____

____

____

____

____

____

____

____

(iv) Write the missing code for the `getAllLoc2Arr(arr2D,value)` function in the `Q1.py` module. This function returns **all** the locations as a list where an item, `value`, appears in the a 2D array, `arr2D`. For example, the following code:

```
arr2=[['a','b','c'],['c','a'],['a'],'a']
print(getAllLoc2Arr(arr2,'a'))
print(getAllLoc2Arr(arr2,'b'))
print(getAllLoc2Arr(arr2,'z'))
```

would display this:

```
[[0, 0], [1, 1], [2, 0], [3, 0]]
[[0, 1]]
[]
```

Write the code below.

[10]

```
def getAllLoc2Arr(arr2D,value):
```

Question 2 - Recursion [17]

Examine the `test3_Q2_2016.py` module listed on the last sheet of the test and answer the following questions.

(i) Write down the **exact output** when this module is executed (e.g. when the user presses the "Run" button in Wing101)? [2]

(ii) Write a recursive version of `test3_Q2_2016.py` [7]

```
def someRec(s):
```

(iii) The recursive approach to calculating Fibonacci numbers (as listed below) is much slower than the iterative approach. Explain why and provide an example to support your explanation. [3]

```
def fibRec(x, n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibRec(n-1)+fibRec(n-2)
```

____

____

____

____

____

____

(iv) Write an iterative version of the recursive Python program below: [5]

```
def strangeRec(s):
    if s == '':
        return []
    else:
        l = strangeRec(s[:-1])
        l.append(ord(s[-1])
        return l
print(strangeRec('Hello'))
```

____

____

____

____

____

____

____

____

____

Code examples for the test – you may detach this sheet.

Question 1

```
#Q1.py

def xWords(wordList):
    if type(wordList)==type({}):
        f1=open("input.txt",'r')
        f2=open("output.txt",'w')
        for line in f1:
            words=line.split()
            for ind in range(len(words)):
                if words[ind] in wordList:
                    words[ind]=wordList[words[ind]]
            print(" ".join(words), file=f2) # join
converts list->string
        f1.close()
        f2.close()


def arrFrmt(values,init):
    result=[]
    for val in values:
        if type(val)==type([]):
            row=[]
            for i in range(val[1]):
                row.append(val[0])
            result.append(row)
        else:
            result.append([0]*val)
    return result


arr1=[1,2,3]
arr2=[['x',2],[1,3],['a',4]]
arr3=[]

x=arrFrmt(arr1,0)
print(x)
x=arrFrmt(arr2,0)
print(x)
x=arrFrmt(arr3,0)
print(x)
```

```
def getAllLoc2Arr(arr2D,value):  #add in missing code
```

Code examples for the test – you may detach this sheet.

Question 2

```
# test3_Q2_2016.py
def someIt(s):
     olist = []
     for i in range(len(s) // 2):
         if s[i] == s[len(s)-1-i]:
             olist.append(s[i])
     return olist


print(someIt('begger'))
print(someIt('X'))
```