University of Cape Town ~~ Department of Computer Science

Computer Science 1015F ~~ 2016

Class Test 3

***** Solutions *****

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		24		16													
Marks	0	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	1	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	2	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	4	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	5	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	6	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	7	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	8	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
	9	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Marker																	

Question 1 – Arrays, Dictionaries and Files [24]

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

(i) From this module, give an example of a:

A. variable that is of type `'list'` [1]

Many examples – result, row, lengths

B. variable that is of type `'dict'` [1]

formats

C. dictionary key [1]

(ii) Describe briefly, and in clear English, what the function `out(arr)` does when called. Your answer must explain what happens with input parameter `arr` of different types. [5]

This function writes to a file called "pattern.txt".[1]
If the `arr` parameter is a 2D array (or list of lists),[1], each row of the array is written as a separate line, [1] with values separated by spaces [1]. Otherwise, no values are written to the file – pattern.txt will be empty.[1]

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

```
[[0], [0, 0], [0, 0, 0]]  #[2]
[['bob', 'bob', 'bob'], ['bob', 'bob', 'bob'], ['bob', 'bob', 'bob']] #[2]
[] #[2]
```

(iv) Write the missing code for the `countWords(filename,word)` function in the `Q1.py` module. This function returns the number of times a given word, `word`, appears in the file named `filename`. You do not need to worry about punctuation or capitalization in your answer (i.e. your function may count "Bob"and "bob" as different words). [10]

```
def countWords(filename,word):
    """Function to count the number of time a given word appears in a file. Case and punctuation sensitive."""
    #fill in code below
    ###remove below from test!!!!
    f=open(filename,'r') #[2]
    count=0 #[1]
    for line in f: #[1]
        words=line.split() #[1]
        for w in words: #[1]
            if w==word:#[1]
                count+=1 #[1]
    f.close() #[1]
    return count #[1]
```

Question 2 - Recursion [16]

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]

```
 eslwziz    # [1] mark
X           # [1] mark
```

(ii) In terms of recursion, what purpose do lines 3-4 and 5-6 serve? [2]

lines 3-4 are the base (or stopping) cases [1]
lines 5-6 is the recursive step [1]

(iii) Consider the effect of replacing line 6 with:

```
return s[-1]+s[0]+someRec(s)
```

What would happen, in practice, if this new program was run in Wing101? [2]

The someRec function would suffer from infinite recursion since the problem size is never reduced [1]. Wing101 would continue recursing until the stack depth (recursion limit) was reached and it would then stop with an error [1].

(iv) Write an iterative version of `test3_Q2_2016.py`. [7]

One correct answer is:

```
def someIt(s):
    y=''   #[1]
    for i in range(len(s)//2): #[1]
        y = y + s[len(s)-i-1]+s[i]  #[2]
    if len(s) % 2 == 1: # add middle character to the end #[1]
        y = y + s[len(s)//2]  #[1]
    return y  #[1]
```

(v) Which version of this program (recursive or iterative) do you think is better and why? [1]

Recursive because it is simpler and arguably easier to understand. [1] Iterative because it is likely to be faster for long strings [1]

(vi) What does the recursive function listed below do? For which inputs will this function fail? [2]

```
def enigmaRec(x, n):
    if n == 1:
        return x
    else:
        return x * enigmaRec(x, n-1)
```

*It calculates x ** n, for n >= 1 [1]. It will fail due to infinite recursion if n < 1. [1]*

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

```
# Module Q1.py
#Q1.py
formats = {"tri":[1,2,3],
           "dmnd":[1,2,4,2,1],
           "sqr":[3,3,3]}

def out(arr2):
    f=open("pattern.txt",'w')
    if type(arr2)==type([]):
        for row in arr2:
            if type(row)==type([]):
                for col in row:
                    print(col,file=f,end=' ')
                print(file=f)
    f.close()

def arrFrmt(val,frmt):
    result=[]
    if frmt in formats:
        lengths=formats[frmt]
        for l in lengths:
            row=[val]*l
            result.append(row)
    return result

def countWords(filename,word):
      """Function to count the number of time a given word
appears in a file. Case and punctuation sensitive."""
    #fill in code

x=arrFrmt(0,"tri")
print(x)
y=arrFrmt('bob',"sqr")
print(y)
y=arrFrmt('x',"hrglss")
print(y)
```

```
#Module test3_Q2_2016.py
def someRec(s):
    if len(s) <= 1:
        return s
    else:
        return s[-1]+s[0]+someRec(s[1:-1])

print(someRec('swizzle'))
print(someRec('X'))
```