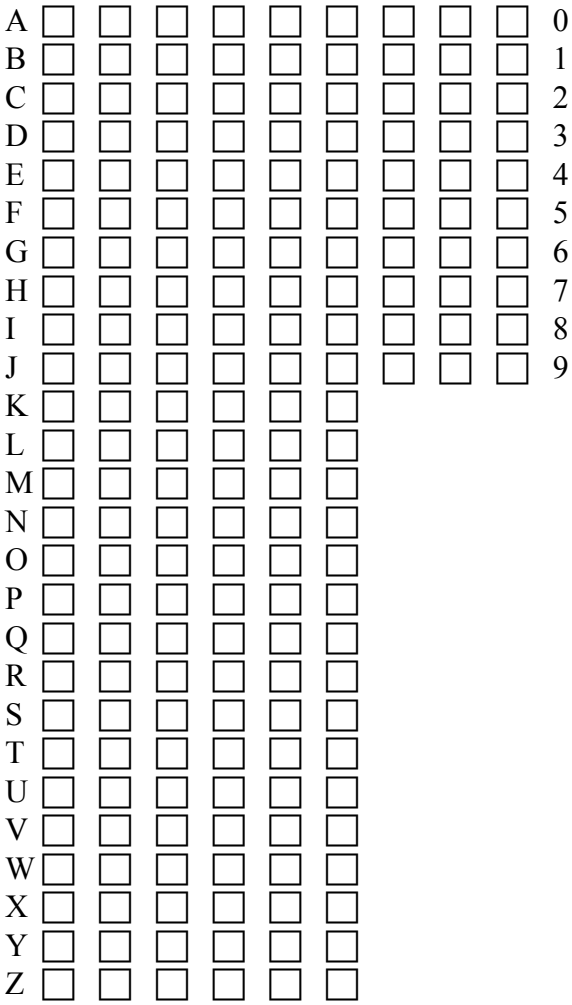
University of Cape Town ~~ Department of Computer Science

Computer Science 1015F ~~ 2015

Class Test 3

** *Solutions* **



Question	1	2	3	4	5	6	7	8
Max	10	9	6	10	15			
Marks	0 1 2 3 4 5 6 7 8 9							
Marker								

Question 1 – Arrays and Dictionaries [30]

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

(a) From this module, give an example of a variable that contains:

i. A list [1]

vals

ii. A two-dimensional array [1]

someData

iii. A dictionary [1]

CourseMarks

(b) Assume that this module has been imported into the Python3 interpreter. For each of the function calls below, write down the value returned, or indicate if an error occurs.

i. `vals.index(2)` [1]

0

ii. `CourseMarks.keys()` [1]

['MAM1000W', 'AST1000F', 'CSC1015F', 'BIO1004F']

iii. `CourseMarks['AST1000F']` [1]

[88, 78, 67, 85]

iv. `CourseMarks[50]` [1]

Error

v. `len(vals)` [1]

3

vi. `len(someData)` [1]

4

vii. `len(someData[1])` [1]

2

viii. `len(CourseMarks)` [1]

4

ix. `loc2D(someData,2)` [2]

[1, 1]

x. `loc2D(someData,20)` [2]

[0, 1]

i. `loc2D(someData,25)` [2]

_____ *[]*

(b) Describe briefly, and in clear English, what the function `puzzle(keys,data)` does. [3]

This function returns a dictionary[1], where the keys are the items in the list val [1]and the values for each key are the number of times that value occurs in the 2D array someData [1]

(c) Write down the value returned by the function call:

`puzzle(vals,someData)` [2]

{2: 1, 100: 1, 10: 3}

(d) Write the code for a new function in the `Q 1` module. This function, `Threshold(arr2D,cutoff)`, returns an array where all items in supplied two-dimensional array that are over the specified cutoff are replaced with the cutoff value. For example (in the Python3 interpreter):

```
>>>Threshold( [[100,4],[20,500]], 30)

[[30, 4], [20, 30]]
```

Complete the code below: [8]

`Threshold(arr2D,cutoff):`

```
def Threshold(arr2D,cutoff):
    for i in range(len(arr2D)): #2
        for j in range(len(arr2D[i])): #2
            if arr2D[i][j]>cutoff:  #2
                arr2D[i][j]=cutoff #1
    return arr2D #1
```

Question 2 – Recursion [20]

(c) Examine the Python3 `Q2.py` module listed at the end of the test and answer the following questions.

(d) What is the base case of the recursive function `MystRec(n)`? [1]

(e) What is the recursive step of the recursive function `MystRec(n)`? [1]

(a) Describe briefly, and in clear English, what the recursive function `MystRec(n)` does. [2]

(b) Write down the exact output of the `Q2.py` module if the user runs the module. [8]

(c) Explain clearly what happens, and why it happens, when a user types the following code in the Python3 interpreter:

```
>>>import Q2
>>>Q2.MystRec(-2)
```

. [2]

(d) Now write the code for a recursive function `FiboRec(n)` that returns the nth Fibonacci number. This function **must be recursive** – no marks will be given for an iterative (non-

recursive) solution. For numbers less than or equal to zero, your function should return the value 0. For example,

```
>>>FiboRec(0)
0
>>>FiboRec(1)
1
>>>FiboRec(2)
1
>>>FiboRec(3)
2
>>>FiboRec(4)
3
```

Complete the code below: [6]

```
def FiboRec(n):
```

def FiboRec(n):
* if n<=0:*
*   return 0 #1 mark*
* if n==1 or n==2: # 2 marks - if statement must not have ' if n==1 or 2 '!*
*   return 1 # 1 amrk*
* return FiboRec(n-1)+FiboRec(n-2) # 2 marks*

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

Question 1

```
# Module Q1.py
someData=[[10,20,30],[1,2],[10,20,40,50],[10,100]]
vals=[2,10,100]

CourseMarks={'CSC1015F':[50,78,34,50],
             'MAM1000W':[35,55,67,90],
             'AST1000F':[88,78,67,85],
             'BIO1004F':[67,68,60,65]}

def loc2D(arr2D,key):
    for i in range(len(arr2D)):
        for j in range(len(arr2D[i])):
            if arr2D[i][j]==key:
                return [i,j]
    return []

def puzzle(keys,data):
    count={}
    for a in keys:
        count[a]=0
        for row in data:
            for i in row:
                if i==a:
                    count[a]=count[a]+1
    return count
```

Question 2

```
# Module Q2.py
def MystRec(n):
    if n==0:
        return 1
    else:
        return MystRec(n-1)*2

w=MystRec(3)
x=MystRec(0)
y=MystRec(8)
z=MystRec(MystRec(2))
print(w)
print(x)
print(y)
print(z)
```