Please fill in your Student Number and Name.

Student Number : _____

Name:

_____

_____

Student Number:

_____

University of Cape Town ~ Department of Computer Science

Computer Science 1015F ~ 2011

Class Test 2

** *SOLUTIONS* **

Question	Max	Mark	Marker
1	22		
2	8		
TOTAL	**30**		

Marks : 30

Time : 40 minutes

Instructions:

a) Answer all questions.

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

c) Show all calculations where applicable.

Question 1 [22]

Examine the module `test2.py` that is listed below:

```
def oddOrEven(a):
    #code missing

def surprise(a,b):
     for i in range(a,b,2):
          print(i,end=' ')
     print()

def secret(a,b,c):
   if a<b:
      if a<c: return a
      else: return c
   elif b<c: return b
   else: return c

def mystery(a,b):
     if a>b  or  a<=0:
          print("Can't run mystery with these values!")
     for i in range(a,b+1):
        count = 0
        for j in range(1,i+1):
            if i%j ==0:
                 count = count + 1
        if count%2 != 0:  print(i)

surprise(3,8)
x=secret(5,2,3)
print(x)
a,b,c = 15,10,8
a = secret(a,b,c)
print(a)
mystery(a,b)
```

(a) **From the module** `test2py,` give an example of:

1. a decision statement [1]

   any of the "if" statements

2. a relational operator [1]

   > , <=, ==, OR !=

3. a loop index variable [1]

   i or j

4. a function call (or invocation) [1]

any functions calls valid, but NOT the definition

range(a,b+1) also valid, as is range(1,i+1) and print(i)

(b) The `oddOrEven` function is missing code. This function should print "Odd number" if the parameter passed to it is odd, and "Even number" if it is even. Add in code below so that the function will work properly. [2]

```
if a%2==0:          [1]
    print("Odd number")     [0.5]
else:
    print("Even number")        [0.5]
```

(c) The `surprise` function uses a definite ('for') loop. Rewrite this function so that it works the same, but now uses an indefinite ('while') loop. [4]

```
i =a   #[1]
while i<b:     #[1]
    print(i,end=' ')     #[0.5]
    i = i+2    #[1]
print() #[0.5]
```

(d) Write down the **exact output** of the `test2.py` module when it is run in the Python interpreter (i.e. when you press "Run" in the Wing IDE). [8]

3 5 7

2

8

9

[2 marks for each line]

(e) Write down a clear, concise description of what the `secret` function does. [2]

The secret function returns [0.5] the smallest of 3 numbers [1] passed as parameters[0.5]

(f) Now write down a clear, concise description of what the `mystery` function does. [2]

The mystery function prints [0.5]

all square numbers[1] OR all numbers with an odd number of factors [1]

within the range provided [0.5] – i.e. in the range from a up to and including b.

Question 2 [8]

Answer the following questions based on the code below that searches through a list of strings for a given word.

```
def search(word):
   the_list = [ 'i', 'love', 'computer', 'science' ]
   position = 1
   while the_list[position]!=word and position<len(the_list):
      position = position + 1
   print (position)
```

(a) What is an **equivalence class**? [1]

set of values for which program is expected/assumed to behave in similar way [1]

(b) Provide a minimal set of test values to support **statement coverage** testing of this function. [1]

any single string – even one not in the list [1]

(c) If we use an exhaustive testing method, we will find that there are 2 errors in this program. Explain what they are; whether each is a syntax error or logic error; and how to fix each error. [6]

1. *position starts at 1 instead of 0 [1], logic error [1], make position=0 in line 3 [1]*
2. *if word is not found print will access element that does not exist [1]; logic error [1]; add if statement before print to check that position is within bounds [1]*