Name:

_____

_____

Student Number:

_____

Please fill in your Student Number and Name.

Student Number : _____

University of Cape Town ~ Department of Computer Science

Computer Science 1015F ~ 2013

June Examination

** *SOLUTIONS* **

Question	Max	Internal	External
1	15		
2	23		
3	8		
4	8		
5	6		
6	5		
TOTAL	**65**		

Marks : 65

Time : 120 minutes

Instructions:

a) Answer all questions.

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

c) Show all calculations where applicable.

Question 1 - Multiple Choice [15]

For each of the multiple choice questions below, **write down the letter** corresponding to the correct answer. Only write down ONE letter. Note that all code examples refer to Python version 3.

i. An algorithm is:

A) always written in unambiguous pseudocode.

B) the best method for calculating a value.

C) the only correct answer to a puzzle or problem.

D) a set of instructions to be followed to solve a problem.

E) All of the above

_____

D

ii. An example of a Python expression is:

A) MOL=42

B) 'asc'

C) 4+5.0

D) B and C

E) A and C

_____

D

iii. `4>3 or ('happy' and 'sad')`

will evaluate in Python3 as:

A) True

B) 'happy'

C) 'sad'

D) 4

_____

C

iv. Unicode is:

A) a binary code for encoding numbers as letters

B) a 7 digit code

C) ASCII

D) a character code for representing letters as numbers

_____

D

v. In Python3, the expression

```
"monsterBlood"[-1:3:-2]
```

will evaluate as:

A) `'doBe'`

B) `'doBes'`

C) `'olr'`

D) `''`

_____

A

vi. On a computer, secondary storage is :

A) volatile

B) read-only

C) modifiable

D) the 'brain' of the computer

E) C and D

_____

C

vii. A compiler is

A) interactive

B) needed every time a program is run

C) one-shot translation

D) very fast

E) C and D

_____

C

viii. `def fn(x):`

In the function definition above, `x` is:

A) An actual parameter

B) A formal parameter

C) A return value

D) A function call

_____

B

ix. `type({})` will evaluate as:

A) NameError

B) <class 'str'>

C) <class 'dict'>

D) <class 'list'>

_____

C

x. To find out whether a particular key (k) was present in a dictionary (d) you would use:

```
A) d.find(k)
B) d.values(k)
C) k in d
D) d.keys(k)
```

_____

C

xi. For the purpose of path testing, how many paths are there in the following Python function?

```
def pontificate(n):
    if n % 2 == 0:
        print("even")
    else
        print("odd")
    if n < 0:
        print("negative")
    elif n == 0:
        print("zero")
```

A) 2

B) 4

C) 5

D) 6

_____

D

xii. If this is typed in at the Python 3 prompt:

```
>>> lst = [1,2,3]
>>> lst.append([4])
>>> lst
```

It will return:

A) [4, 1, 2, 3]

B) [1, 2, 3, 4]

C) [1, 2, 3, [4]]

D) {1, 2, 3, 4}

_____

C

xiii. The worst case Big-O efficiency of Quick Sort is:

A) O(log n)

B) O(n)

C) O(n log n)

D) $O(n^2)$

_____

D

xiv. Which of the following generally produces the least number of test cases:

A) Exhaustive Testing

B) Statement Coverage

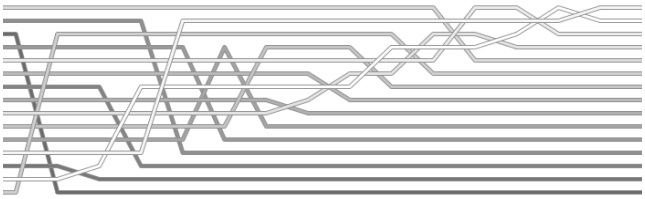
C) Path Testing

D) Boundary Value Analysis

_____

B

xv. Which sorting algorithm does this image depict?



A) Selection Sort

B) Merge Sort

C) Quick Sort

D) None of the above

_____

A

Question 2 [23]

Examine the **Q2.py** module listed on the last sheet of the examination and answer the following questions.

i. From the module **Q2.py ,** give an example of:

A) floor division [1]

_____

a//2 or b//2

B) a string function [1]

_____

find

ii. Explain why you should not replace the simultaneous assignment statement

```
a,b = a/2+b/2, a//2+b//2
```

used in function `convert` with the statements

```
a=a/2+b/2
b=a//2+b//2
```

[1]

_____

_____

_____

_____

In the second option will change the answer, as the value of a will be overwritten. You would need a temporary variable to avoid this.

iii. Why is it necessary for a dynamically-typed language like Python to have a `type` function (as used in module Q2.py) ? [2]

_____

_____

_____

_____

_____

The type function gives the current data type of a variable. In a dynamically-typed language variables can change type. If you want to find out the current data type of a variable , you need a type function.[2]

iv. Can you replace either instance of the boolean operator **`and`** in the function `convert` with an **`or`** operator? Explain your answer. [2]

no, you can't [1], because this would result in an error if you send mixed types for the function (e.g. a string or an integer) – which would make the statement true[1] – you can't do slicing on integers or division on strings.

v. Explain how the builtin Python **`max`** function used in `swivel` uses Unicode in order to calculate an answer. [2]

The max function as used in swivel returns the "biggest" letter in a string. The biggest letter is calculated according to the letter with the highest Unicode value → which is assigned according to lexographic ordering.[2]

vi. Write down the exact output of the Q2.py module when it is run in the Python3 interpreter (i.e. when you press "Run" in the Wing IDE). [6]

('pear', 'apples') #[2] – won't be too fussy about it being a tuple

opples | arange #[2]

*8.5***8.0 #[2]*

vii. Assume that your are now not allowed to use the string slicing operator (:). Now rewrite the `swivel` function so that it works exactly the same as before, but does not use string slicing. You may, of course, index a string position - "a[0]" - just not a range – "a[1:5]". [8]

```
def swivel(a,b):
    if len(a)==len(b): #[1]
        spltV=max(a)
        spltPT=a.find(spltV)
        i=0 # [1]
        tmpA,tmpB="","" #[1]
        while i<len(a): #[1]
            if i<spltPT:  #[1]
                tmpA+=a[i]
                tmpB+=b[i]  #[1]
            else :
                tmpA+=b[i] #[1]
                tmpB+=a[i]
            i+=1 #[1]
    return(tmpA,tmpB)
```

Question 3[8]

i. Write a function `chop(values, max)` that will take in a list of numbers (called `values`) and a threshold (called `max`). It must iteratively chop off both the front and end of the list until the sum of the remaining elements is less than max and return the resulting list. It is important that your function not change the `values` list.

For example:

```
>>> chopped1 = chop([1,2,3,4], 9)
>>> chopped1
>>> [2,3]
>>> chopped2 = chop([5,2,7,4,9], 11)
>>> chopped2
>>> [7]
```

[8]

def chop(values, max):

sum = max+1 *[1]*

```
chopped = values[:]                                    [1]
while sum >= max:                                      [1]
    sum = 0                                            [1/2]
    for i in range(len(chopped)):                      [1]
        sum += chopped[i]                              [1/2]
    if sum >= max:                                     [1]
        chopped = chopped[1:-1]                        [1]
return chopped                                         [1]
```

Question 4 [8]

Examine the **Q5.py** module listed on the last sheet of the examination and answer the following questions.

ii. Explain, in general terms, what the function `recurse(n,2)` does. [2]

It determines recursively whether the number n is a prime (returning True) or not (returning False) [1]. It does this by checking whether n is exactly divisible by any of the numbers in the range 2, ..., n-1. [1]

iii. Write down the exact output of this module when it is run in the Python interpreter (i.e. when you press "Run" in the Wing IDE). [2]

True [1]

False [1]

iv. The function `recurse(n,d)` uses recursion. Rewrite the function as `iterate(n)` so that it produces the same output, but uses iteration. [4]

_____

_____

_____

_____

```
def iterate(n):
    prime = True [1]
    for d in range(2,n): [1]
        if n % d == 0: [1]
            prime = False [1]
            break [1]
     return prime [1]
```

Question 5 [6]

Given a list of numbers as follows:

10	12	10	8	6	4	2

i. Show how the Merge Sort algorithm can be applied to this list to sort the items in ascending order. Show each step of the algorithm, including both the split and merge stages. [4]

10,12,10 8,6,4,2 [1/2]

10 12,10 8,6 4,2 [1/2]

10 12 10 8 6 4 2 [1/2]

10 10,12 6,8 2,4 [1]

10,10,12 2,4,6,8 [1]

2,4,6,8,10,10,12 [1/2]

ii. Do you think it would be faster to apply a quick sort to this list? Explain your answer [2]

_____

_____

No, quicksort will likely be slower [1]

A reversed list is one of the worst cases for quick sort, for which it is O(n^2) [1]

Question 6 [5]

i. Consider this code:

```
1   num = int(input("Enter an integer (-1 to end): "))
2
3   while num != -1:
4
5         print("Remainder of 10 / ", num, " equals ", 10 % num)
6
7         num = int(input("Enter an integer (-1 to end): "))
```

After running it with a certain input it halts, giving this error:

```
Traceback (most recent call last):
 File "divexcep.py", line 3, in <module>
  print("Remainder of 10 / ", num, " equals ", 10 % num)
ZeroDivisionError: integer division or modulo by zero
```

Use Python's exception mechanism to fix this problem.

[2]

_____

_____

_____

_____

_____

_____

_____

num = int(input("Enter an integer (-1 to end): "))
while num != -1:
try: ## Model answer code
print("Remainder of 10 / ", num, " equals ", 10 % num)
except ZeroDivisionError: # Model answer code
print("Can't divide by zero.") # Model answer code
num = int(input("Enter an integer (-1 to end): "))

Note: -1 if the specific exception is not catered for.

ii. Find the value represented by the floating point number written below using IEEE754 representation (bias of 127 added to the exponent; first bit is the sign bit, next 8 bits are the biased exponent, rightmost 23 bits are the significand). Show all your working.

```
1 10000011 11010000000000000000000
```

[3]

_____

_____

_____

_____

_____

_____

_____

Sign is negative. So this is a negative number (0.5 marks)

Biased exponent is 128 + 2 + 1 = 131 (0.5 marks)

Exponent is 131 - 127 = 4 (0.5 marks)

Significand is 0.1101 = 1/2 + 1/4 + 0 + 1/16 = 0.8125 (0.5 marks)

*Result is -1 * (1 + 0.8125) * 2 ^ 4 (0.5 marks)*

Answer is -29 (0.5 marks)

Code examples – you may detach this sheet.

Question 2

```
#Q2.py
def convert(a,b):
    if type(a)==str and type(b)==str:
        return swivel(a,b)
    elif type(a)==int and type(b)==int:
        a,b = a/2+b/2, a//2+b//2
    else:
        a,b = b,a
    return(a,b)

def swivel(a,b):
    if len(a)==len(b):
        spltV=max(a)
        spltPT=a.find(spltV)
        a,b=a[:spltPT]+b[spltPT:],b[:spltPT]+a[spltPT:]
    return(a,b)

y= convert("pear","apples")
print(y)
y= convert("orange","apples")
print(y[0], '|', y[1])
y= convert(5,12)
print(y[0], y[1],sep='***')
```

Question 5

```
#Q5.py
def recurse(n, d):
    if d >= n:
        return True
    elif n % d == 0:
        return False
    else:
        return recurse(n, d+1)

def driver(n):
    return recurse(n, 2)

print(driver(11))
print(driver(9))
```