Please fill in your Student Number and Name.

Student Number : ________________________________

Name:

____

____

Student Number:

____

University of Cape Town ~ Department of Computer Science

Computer Science 1015F ~ 2016

June Examination

*** SOLUTIONS ***

Question	Max	Internal	External
1	10		
2	20		
3	10		
4	10		
5	10		
6	10		
TOTAL	**70**		

Marks : 70

Time : 120 minutes + 10 minutes reading time at the start

Instructions:

a) Answer all questions.

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

c) You may use a calculator, BUT show all calculations where required.

Question 1 [10]

Examine this program and answer the questions that follow.

```
def one(s):
    x = 'code:'
    for i in range (len (s)):
        y = ord(s[i].lower())
        for j in range((y - ord('a')) + 1):
            x += s[i]

    return x

def two(s):
    x=''
    for i in range(len (s)):
        if (i+1) % 2 == 1:
            x += (i+1) * s[i]
        else:
            x += s[i]
    return x
```

For each question below, **write down ONE letter** corresponding to the correct answer.

(a) What is an example of an identifier?

a. `1`

b. `+=`

c. `:`

d. `'code:'`

e. `one`

____

(b) What is an example of a string literal?

a. `1`

b. `+=`

c. `:`

d. `'code:'`

e. `one`

____

(c) What numbers are generated by `range(3,9,2)`?

a. 3,4,5,6,7,8,9

b. 4,5,6,7,8

c. 3,5,7

d. 3,5,7,9

e. 4,6,8

____

C

(d) What string is generated by `'csc1015'[5:2:-2]`?

a. ''

b. '11'

c. '11s'

d. '0c'

e. '0cc'

____

B

(e) What is an advantage of using assembler over an interpreted programming language like Python?

a. shorter code

b. executes faster

c. matches machine code

d. a and b

e. b and c

____

E

(f) Which storage can NOT be changed?

a. RAM
b. flash drive
c. hard drive
d. ROM
e. CPU cache

____

D

(g) What is returned by `one('abc')`?

a. 'code:abc'
b. 'code:aabbcc'
c. 'code:aaabbbccc'
d. 'code:abbccc'
e. 'ABC'

____

f. D

(h) What is returned by `one('DAD')`?

a. 'code:ddaadd'
b. 'code:ddddaadddd'
c. 'code:DDAADD'
d. 'code:DDDAAADDD'
e. 'code:DDDDADDDD'

____

E

(i) What is returned by `two('abcd')`?

a. 'abcccd'
b. 'abbcccdddd'
c. 'aaabccccd'
d. 'aabbbccccdddddd'
e. 'abbcdddd'

____

A

(j) What is returned by `one(two('abc'))`?

a. 'code:aaabbbccc'
b. 'code:abbbbccc'
c. 'code:abbccccccccc'
d. 'code:abbccc'
e. 'code:aabbbccccc'

____

C

Question 2 [20]

Examine this program and answer the questions that follow.

```
def three (a):
    for b in range (a, 0, -1):
        for c in range (2*(a-b)+1):
            print (" ",end="")
        for c in range (2*b-1):
            print ("*",end="")
        print ()

three (eval( input()))
```

(a) Explain in English and at a high level what the `three()` function does. [2]

Draws an inverted triangle [2]

(b) What is the output from `three(-2)`? [1]

Nothing [1]

(c) What is the output from `three(4)`? [2]

```
*******
  *****
    ***
      *
```

[2]

(d) Provide meaningful names for the variables `a, b, c`. [3]

(e) What does `end=''` do in the `print()` statement? [2]

(f) State precisely what you need to change in the code so that the shape is flipped horizontally. [2]

Remove:

```
for c in range (2*(a-b)+1):
        print (" ",end="")
```

[2]

(g) Rewrite the program to accomplish the same task using only a single loop. [4]

(code or pseudo code is fine)

```
def three (a):
  for b in range (a, 0, -1):
    print (" "*2*(a-b),end="")
    print ("*"*(2*b-1),end="")
    print ()

three (eval( input()))
```

[4]

(h) Rewrite the following statement using a `for` loop instead of a `while` loop. [2]

```
while n > 0:
    sum = sum + n
    n = n - 1
```

```
for i in range(n,0,-1):
    sum = sum + I
```

[2]

(i) Write the code to replace/encapsulate the final line of code in the supplied program such that this program may be used as a module or a standalone program. [2]

```
def main ():
   three (eval (input ()))
if __name__="__main__":
   main ()
```

Question 3 [10]

For each question below, **write down ONE letter** corresponding to the correct answer.

(a) Given the Python statement:

```
data = [ [2, 4, 7], [3, 5, 8], [9, 6, 1] ]
```

What output is generated by

```
print(data[2][:-1])
```

a. `[3, 5, 8]`

b. `[9, 6]`

c. `1`

d. `[5, 8]`

e. `6`

____

b.

(b) What does it mean if a sorting algorithm is *stable*?

a. It uses very little additional memory

b. It is very fast relative to other sorting algorithms

c. It maintains the relative order of items with equal values

d. It does less work if the list is already sorted

e. It sorts the list in reverse order

____

c.

(c) Given the following list:

```
data = ['first', 'second', 'third', 'fourth', 'fifth']
```

What output is generated by:

```
print(data[5])
```

a. `first`

b. `fifth`

c. `['first', 'second', 'third', 'fourth', 'fifth']`

d. `an IndexError`

e. `'fifth'`

d.

____

(d) What is the worst case-complexity of the Merge Sort algorithm?

a. $O(n^2)$

b. O(n log n)

c. O(n)

d. O(log n)

e. O(1)

____

b.

(e) Which of these Python commands will NOT write the string “Hello World” to a file, f ?

a. `f.write('Hello World')`

b. `print('Hello World', file=f)`

c. `f.writelines(['Hello World'])`

d. `print('Hello World')`

____

d.

(f) Which of the following is a glass box testing strategy?

a. debugging

b. tracing

c. statement coverage

d. random testing

e. all of the above

____

C

(g) What is the time complexity of the **binary search** algorithm in the **worst case**?

a. O(1)

b. O(log n)

c. O(n log n)

d. O(n^2)

e. O(n)

B

(h) Consider

```
def myst(n):
    return 3*myst(n-1)
```

The function `myst(n)` calculates:

a. The nth Fibonnacci number

b. n!

c. 3^n

d. n^m

e. nothing (infinite recursion)

E

(i) 76_8 is equal to:

a. 90_{10}

b. 32_2

c. $1A_{16}$

d. All of the above

e. None of the above

e

(j) What is the Boolean expression *F* represented by the following truth table?

A	**B**	**C**	***F***
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

a. A OR B

b. **B** AND **C**

c. A NAND B

d. **A** AND **B** AND **C** OR **A**

e. **A** AND **B** OR **B** AND **C**

e

Question 4 [10]

Examine the `exam_Q4.py` module listed on the last sheet of the exam and the input file `exam_Q4_input.txt` and answer the following questions.

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

____

____

____

____

none [1]

None [1]

(b) Rewrite the `process` and `search` functions so that they create and use a Dictionary instead of a List. [3]

```
def process(inData, dictData):
```

____

____

____

____

____

____

____

____

```
def search(dictData):
```

____

____

____

____

____

____

____

____

One correct answer is:

```
def process(inData, dictData):
   for i in range(len(inData)):
        inData[i] = inData[i].replace('\n', '')
        inData[i] = inData[i].split(',')
        dictData[inData[i][0]] = inData[i][1]      #[1]

def search(dictData, term):
    if term in dictData:   #[1]
        return dictData[term]    #[1]
```

(c) What is the advantage of using a dictionary instead of a list for this problem? [1]

Searching for an item associated with a particular key is much faster for a dictionary (O(1)) than an unordered list (O(n)) [1]

(d) Rewrite the `load` function so that it handles the situation where `exam_Q4_input.txt` does not exist. by asking the user for a new file name and loading that file instead. This should repeat until a valid file is loaded. [3]

```
def load():
```

One correct answer is:

```
fin = False #[½]
filename = 'exam_q4_input.txt'

while not fin: # [½]
    try: # [½]
        dataFile = open(filename, 'r', encoding='utf-8')
        data = dataFile.readlines()
        dataFile.close()
        fin = True # [½]
    except IOError: # [½]
        filename = input("enter a new file:")  # [½]
return data
```

(e) Why is the process data function not written as follows? [1]

```
def process(inData):

    for line in inData:
        line = line.replace('\n', '')
        line = line.split(',')
```

This function does not actually change the data inside inData but rather the line variable instead. [1]

Question 5 [10]

Examine the Python code below and answer the questions that follow.

```
def riddle(n):
    if n<=0:
        return 0
    a=1
    b=1
    for i in range(n-2):
        a,b=a+b,a
    return a
```

(a) You are asked to test the `riddle` function using a range of testing strategies.

a. Which would be better, random testing or exhaustive testing? Explain your answer. [1]

b. Write down a **minimal** set of inputs to this function that would constitute a complete **statement test.** [1]

Two values required: any integer>=2, an integer <=0

c. Write down a **minimal** set of inputs to this function that would constitute a complete **path test.** [1]

`Three values required: An integer >=2, an integer <=0, 1 or 2

(b) Rewrite the `riddle` function below to use recursion instead of iteration. [4]

```
def riddle_rec(n):
```

```
def riddle_rec(n):
  if n<=0: return 0 #[1]
  if n==1 or n==2:
    return 1 #[1]
  return riddle_rec(n-1)+riddle_rec(n-2) #[2]
```

(c) Calculate 7_{10}-17_{10} using 8-bit 2's complement binary addition. Show all calculations. [3]

7_{10}-17_{10}

= 00000111 + 2's complement (0010001) [1]

=00000111 + 11101110[1]

=11101111 [1]

Question 6 [10]

The generation of complex computer graphics requires a lot of mathematical calculation, sometimes including trigonometic functions like sine. This can be very slow, because trigonometric functions are inherently slow to compute. To speed this up, we can use a mechanism called a **cache**, where we temporarily store recent answers so we can reuse them without recalculating them.

Your task is to write a function, as defined below, to serve as a cache.

sine_cache (x, cache)

- *x* - the value to calculate the sine of.
- *cache* - an array/list of at most 10 items that stores previous values. Each entry in the cache is of the form *[x, sine_x, time]*, where *sine_x* is the sine value for *x* and *time* is when the value was last calculated.

Your function must first check if *x* exists in the cache. If it does, the associated *sine_x* value must be returned. If *x* does not exist, invoke the function **math.sin(x)** and return this value, but also store these values in an empty position in the array. If the cache array already has 10 items, replace the entry with the lowest time value.

When storing entries into the cache, you can get the current time as a number of seconds using the function **time()** from the **time library**.

You MAY NOT use a dictionary or any other data structure for the cache. [10]

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

```
def sine_cache (x, cache):
    for entry in cache:    [lookup - 3]
        if entry[0]==x:
            return entry[1]
    sine_x = math.sin (x)    [actual calculation and returning value - 1]
    if len (cache)==10:     [adding to full array - 4]
        min = cache[0][2]
        for i in range (1, 10):
            if cache[i][2] < min:
                min = i
        cache[i] = [x, sine_x, time.time()]
    else:
        cache.append ([x, sine_x, time.time()])    [adding to array - 2]
    return sine_x
```

Code examples for the examination (you may detach this sheet).

__________________Question 4_________________________

```
#exam_q4.py

  def search(data, term):
      for row in data:
          if row[0] == term:
              return row[1]

  def process(inData):
      for i in range(len(inData)):
          inData[i] = inData[i].replace('\n', '')
          inData[i] = inData[i].split(',')

  def load():
      dataFile = open('exam_q4_input.txt', 'r')
      data = dataFile.readlines()
      dataFile.close()
      return data

  data = load()
  process(data)
  print(search(data, 'A'))
  print(search(data, 'lock'))
```

```
  # exam_q4_input.txt

  Hepburn,12
  Day-Lewis,5
  Streep,19
  Nicholson,12
  Washington,6
  Brando,8
```