Please fill in your Student Number and Name.

Student Number : _____

Name:

_____

_____

Student Number:

_____

University of Cape Town ~ Department of Computer Science

Computer Science 1015F ~ 2015

June Examination

** *SOLUTIONS* **

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

Marks : 70

Time : 120 minutes

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.

```
# note: 97 is the Unicode value of 'a' and
#       65 is the Unicode value of 'A'

def one (s):
    x = ''
    for i in range (len (s)):
        if i % 2 == 1:
            x += chr (ord(s[i]) - 32)
        else:
            x += s[i]
    return x

def two (s):
    x = ''
    while len(s)>0:
        x = s[0] + x
        s = s[1:]
    return x
```

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

(a) What is an example of an identifier?

a. one
b. 2
c. []
d. "
e. :

_____

A

(b) What is an example of an integer literal?

a. one
b. 2
c. []
d. "
e. :

_____

B

(c) What numbers are generated by range (2,5)?

a. 2,3,4,5
b. 3,4
c. 2,3,4
d. 3,4,5
e. 2,2,2,2,2

_____

C

(d) What string is generated by 'hello'[3:1:-1]?

a. "
b. 'ell'
c. 'll'
d. 'lle'
e. 'olleh'

_____

C

(e) What are some advantages of interpreted programming languages like Python over compiled languages like C++?

a. faster execution

b. shorter code

c. no compile step

d. a and b

e. b and c

_____

E

(f) Storage that loses its contents when computer is powered down.

a. CPU cache

b. flash drive

c. hard drive

d. RAM

e. ROM

_____

D

(g) What is returned by one ('abcdefg')?

a. abcdefg

b. gfedcba

c. ABcdEFg

d. gFeDcBa

e. aBcDeFg

_____

E

(h) What is returned by two ('abcdefg')?

a. abcdefg

b. gfedcba

c. ABcdEFg

d. gFeDcBa

e. aBcDeFg

_____

B

(i) What is returned by two (one ('yellow'))?

a. yElLoW

b. wOlLeY

c. YeLlOw

d. YELLOW

e. WoLlEy

_____

E

(j) What is returned by one (two ('yellow'))?

a. yElLoW

b. wOlLeY

c. YeLlOw

d. YELLOW

e. WoLlEy

_____

B

Question 2 [20]

Examine this program and answer the questions that follow.

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

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

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

draws an isosceles triangle [2]

(b) What is the output from `three (3)`? [2]

```
  *
 ***
*****
```
[2]

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

(nothing) [1]

(d) Provide a set of more meaningful variable names for `x`, `a` and `b`. [3]

x = height [1]

a = row [1]

b = column [1]

(e) What does the `print` statement without any arguments do? [2]

it moves the cursor to the next line [2] or it is equivalent to a carriage return/newline [2]

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

change the first range to (x, 0, -1) [2]

(g) Explain how you can change this code so it will work either as a standalone program or a reusable module. [2]

add if __name__ == "__main__": just before the last line and indent the last line [2]

(h) State an alternative algorithm to accomplish the same task without explicit nested for/while loops. [3]

*create x as a string s of 2*h *s*

create y as a string of h spaces

iterate i from 1 to height

  print a substring of length i-1 from y

  *print a substring of length 2*i-1 from x*

(i) Rewrite this statement so it has the same effect but uses a `while` loop instead of a `for` loop.[3]

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

b = 0 [1]

*while b<2*a-1: [1]*

  print ("", end="")*

  b += 1 [1]

Question 3 [10]

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

(a) Which of these is a runtime error in Python?

a. incorrect indentation

b. adding two strings together

c. misspelling a keyword

d. division by zero

e. all of the above

_____

D

(b) 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

(c) If `X=[[10,1],[[20,'exam',30], 20, [3,9,50]]]`
then `len(X)` is

a. 1

b. 4

c. 3

d. 10

e. 2

_____

(d) If `X=[[10,1],[[20,'exam',30], 20, [3,9,50]]]`
then `X[1][2][1]` is

a. 1

b. 'exam'

c. builtins.TypeError

d. 3

e. 9

_____

E

(e)

```
def myst(n,m):
  if m<1: return 1
  return myst(n,m-1)*n
```

The function `myst(n)` calculates

a. The nth Fibonnacci number

b. n!

c. m^n

d. n^m

e. nothing (infinite recursion)

_____

D

(f) 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

(g) Which sorting algorithm has the best time efficiency in the worst case?

a. Quicksort

b. Merge sort

c. Selection sort

d. Insertion sort

_____

B

(h) What is the floating point value of the following IEEE 754 single precision number?
1 10000000 01100000000000000000000

a. -2.75

b. 2.5

c. 5.5

d. -5.5

e. -2.125

_____

A

(i) $1A_{16}$ is equal to

a. 240_{10}

b. 32_8

c. $0001\ 1111_2$

d. All of the above

e. None of the above

_____

B

(j) What is the Boolean operation *F* represented by this truth table?

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

a. A OR B

b. A AND B

c. A NAND B

d. A XOR B

e. NOT (A AND B)

_____

D

Question 4 [20]

Examine the `Q4.py` module listed at the end of this examination and answer the following questions.

(a) Suppose that you want to test the `triangle` function by itself using the path coverage testing technique. Write down a minimal list of input values that will constitute path coverage testing of this function. [2]

[] (empty list)

[10,20] (list with one or more elements)

(b) Write down the exact output of the `Q4.py` module when it is run in the Python3 interpreter. [3]

20

20 30

20 30 10

20 30 10 40

20 30 10 40 12

#tests understanding of the multidimensional arrays and the code

(c) Write the `triangle` function using recursion. [4]

def triangle(list):

```
if list:
    new_list=triangle(list[0:-1]) #1
    new_list.append(list)#1
    return new_list #1
return [] # 1 mark for stopping case
```

(d) Write Python code to write the contents of `new_list` to the text file named `Tri.txt`, with one value per line in the file and no empty lines. [4]

```
f=open("Tri.txt",'w') #1 mark
for i in range(len(new_list)): #1 mark
    for j in range(len(new_list[i])): #1 mark
        print(new_list[i][j],file=f) #1 mark
f.close()
```

(e) Show the steps of the merge sort algorithm when applied to sort `new_list`. [2]

```
[20,30,10,40,12,90,70,80]
[20,30,10,40],[12,90,70,80]
[20,30,][10,40],[12,90][70,80]
[20,][30,][10,][40],[12,][90][70,][80]
[20,30,][10,40],[12,90][70,80]
[10,20,30,40],[12,70,80,90]
[10,12,20,30,40,70,80,90]
```

(f) Convert the decimal number 95_{10} to octal using repeat division. Show all calculations. [2]

95/8 = 11 r 7

11/8= 1 r 3

1/8 = 0 r 1

=137_8

(g) Calculate 8_{10}-13_{10} using 8-bit 1's complement binary addition. Show all calculations. [3]

8_{10}-13_{10}

= 00001000 + 1's complement (00001101) [1]

=00001000 + 11110010 [1]

=1111010

=complement(00000101) [1]

=-5

Question 5 [10]

Alternate Statistics

At the end of the semester, the Computer Science department compares CSC1015 marks to those for previous years to check for quality, trends in performance, etc. Such comparisons can be made on the basis of simple statistics, like the pass rate or the average mark. These statistics are sometimes controversial, as they do not fully capture the performance of a highly varied class of students.

As one candidate for an alternate statistic, the department would like to also calculate the average of the lowest 5 marks.

Write a Python program where the user can input an unsorted list of marks, at the end of which the program will output the average of the lowest 5 marks. The user must enter marks one at a time, ending with a value of -1.

Your program must work for any number of marks from 1-1000. Assume that there are no errors during input and do not round/truncate your answer when printing out the result.

Examples:

- If the user enters the numbers 1, 3, 5, 7, 9, 8, 6, 4, and 2, the output value is 3.0.
- If the user enters the numbers 1, 3, 4, and 2, the output value is 2.5.

```
marks = []
score = int (input ("Enter a score:"))
while score != -1:
    marks.append (score)
    score = int (input ("Enter a score:"))
marks.sort ()
sum = 0
for i in range (min (len (marks), 5)):
    sum += marks[i]
print ("Average = ",sum/min (len(marks), 5))
```

input and storage [3]

sorting [1]

average calculation, with correct cutoff [4]

loops and general structure [2]Code examples for the test – you may detach this sheet.

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

Question 4

```
#Module Q4.py
def triangle(lst):
    width=0
    output=[]
    for i in range(len(lst)):
        row=[]
        for j in range(i+1):
            row.append(lst[j])
        output.append(row)
    return output

test_list=[20,30,10,40,12]
new_list=triangle(test_list)

for i in range(len(new_list)):
    for j in range(len(new_list[i])):
        print(new_list[i][j],end=' ')
    print()
```