

UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

Department of Computing

B. Sc. Examination 2017

IS51021B Problem Solving For Computer Science

Duration: 2 hours 15 minutes

Date and time:

This paper is in two parts: part A and part B.

You should answer ALL questions from part A and TWO questions from part B.

Part A carries 40 marks, and each question from part B carries 30 marks.

The marks for each part of a question are indicated at the end of the part in [.] brackets.

There are 100 marks available on this paper.

You are not allowed to use any electronic device (such as mobile telephones, calculators, laptops, tablets) during the exam.

**THIS PAPER MUST NOT BE REMOVED
FROM THE EXAMINATION ROOM**

Part A

You should attempt all of these questions

Each question has one correct answer.

(a) What does execution of the following code fragment produce on the screen:

```
a = 0
b = 2
c = 0
print('(a/b)/(1/c) =', float(a/b)*c)
```

- i. $(a/b)/(1/c) =$
- ii. $(a/b)/(1/c) = 0$
- iii. $(a/b)/(1/c) = 0.0$
- iv. A syntax error
- v. A run-time error
- vi. None of the above

[5]

(b) What is the final value of variable *c* after the following statements have been executed:

```
(a,b,c) = (0,-1,2)
b *= 2
c = a-b-c
(a,b,c) = (a,c,b)
```

- i. 0
- ii. 1
- iii. -1
- iv. 2
- v. -2
- vi. None of the above

[5]

(c) Consider the following code fragment:

```
import random
x = random.randrange(0,2)
```

After its execution, the variable *x* will contain:

- i. A (pseudo)random integer *N* such that $0 \leq N < 2$
- ii. A (pseudo)random integer *N* such that $0 \leq N < 1$
- iii. A (pseudo)random floating point number in the range [0.0, 2.0)
- iv. A (pseudo)random floating point number in the range [0.0, 1.0)
- v. None of the above

[5]

(d) What does execution of the following code fragment produce on the screen:

```
x = 0
if x - 2 < -1:
    print(2 ** 3 == 7)
else:
    print(not x - 2 < -1)
```

- i. An error
- ii. "2 ** 3 == 7"
- iii. "not x - 2 < -1"
- iv. False
- v. "True"

[5]

(e) Which sequence will the following code produce in output (assume the numbers separated by commas are printed on a different lines):

```
x = -2
y = -1
for i in range(4, y, x):
    print(i)
```

- i. 4, 2, 0, -2
- ii. 4, 2, 0
- iii. 4, 2
- iv. 2, 0
- v. 2
- vi. None of the above.

[5]

(f) What will be printed on the screen as a result of executing the following lines of code?

```
def div(x, y):
    print(y%x)

result = div(2, 5)
print(result)
```

- i. 1
- ii. 2
- iii. 5
- iv. An error message
- v. None of the above

[5]

(g) Which message will be printed on the screen by executing the following commands:

```
s = "Point " + str(3.50) + ") 'Tab' is '\t', and not '/t'"
print(s)
```

- i. Point 3.50) Tab is '\t', and not ' /t'
- ii. Point 3.5) Tab is ' /t', and not '\t'
- iii. Point 3.50) Tab is '\t', and not '/t'
- iv. Point 3.5) Tab is ' /t', and not '/t'
- v. Point 3.5) Tab is '\t', and not '/t'
- vi. Point 3.5) Tab is ' /t', and not '/t'

[5]

(h) What does execution of the code extract below produce?

```
b = []
a = [[1,2,3], True, [1,2,3]]
b = a
b.append("False")
print(a[3][1])
```

- i. An error message
- ii. 1
- iii. 2
- iv. 3
- v. F
- vi. a
- vii. "False"

[5]

Part B

You should attempt two of these three questions

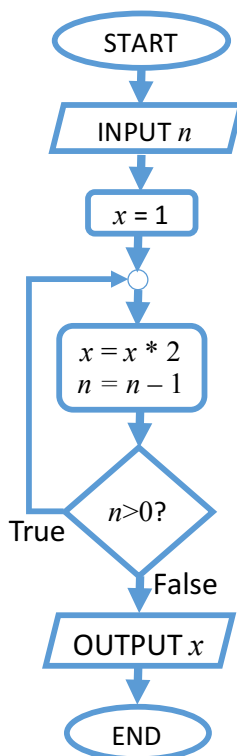
Question B1

(a) What is the output of the following program? Justify your answer with an explanation.

```
def swap(x,y):  
    temp = x  
    x = y  
    y = x  
  
def main(b,a):  
    swap(a,b)  
    print(a,b)  
  
a = 2  
b = 3  
main(a,b)
```

[10]

(b) Consider the algorithm specified by the following flowchart (assume n integer and $n > 0$):



i). Which value does the algorithm print in output for any given input $n > 0$?

[4]

ii). Write a Python implementation of the algorithm which uses a **'for' loop** to compute the same result.

[8]

iii). Write a Python implementation of the algorithm which uses a **'while' loop** to calculate the same result.

[8]

Question B2

(a)

- i). What is the result returned by the following Python function do? Assume the input parameter x is a string.

```
def secret(x):
    i = len(x)-1
    while i >= 0:
        x = x + x[i]
        i -= 1
    return x
```

[6]

- ii). Will the above function work also if x is a list of integers? Justify your answer.

[8]

- (b) Explain how the “bubblesort” algorithm works. Provide an example involving sorting a list of integers.

[8]

- (c) Write a recursive function that accepts as input a (possibly empty) list of numbers and returns the sum of the squares of all the numbers contained in the list.

For example, if the list given is $[1, 3.5, -0.7]$, the value returned should be $(1)^2 + (3.5)^2 + (-0.7)^2$, i.e., 13.74

[8]

Question B3

- (a) Explain what happens to the internal representation of names and objects that Python maintains when the following statements are executed:

i).

```
tess = turtle.Turtle()
john = turtle.Turtle()
```

[3]

ii).

```
tess = john
```

[3]

iii).

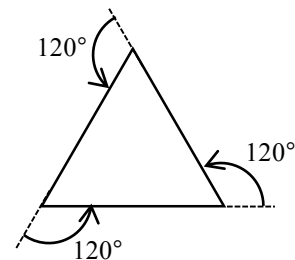
```
import turtle
```

[4]

- (b) Consider the following program for drawing an equilateral triangle having sides of 3 different colours (refer also to the figure on the right, where the triangle's exterior angles are highlighted):

```
import turtle
def drawIt(t):
    for i in [0,1,2]:
        t.color(colrs[i])
        t.forward(100)
        t.left(120)

w = turtle.Screen()
tess = turtle.Turtle()
colrs = ['red', 'blue', 'green']
drawIt(tess)
```



Modify the function “drawIt ()” to make it more abstract and generic, so that it can be used to draw:

- i). an equilateral triangle having arbitrary side length x (i.e., x should be an additional input parameter of the revised function drawIt ());

[2]

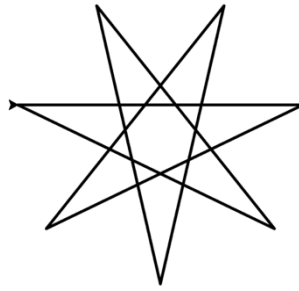
- ii). an equilateral triangle having side of arbitrary length x as above, and such that each side is assigned a color chosen at random amongst red, blue and green;

[3]

- iii). any regular polygon of n sides, each of length x and color chosen at random as in point ii). above. Side length x and exterior angle a (in degrees) should be passed as parameters to the revised function, drawIt (). (Recall that the exterior angles of a regular polygon always add up to 360°).

[5]

(c). Write a program that draws the figure shown below (assume each segment is 100 pixels long):



[10]

End of Exam