

COMP1021
Introduction to Computer Science

Midterm Examination

Thursday, 28 March 2019
7pm – 8pm
(1 hour duration)

Please note you cannot leave the examination room before the 1 hour has finished.

Your full name:	
Your student id:	
Your lecture: <i>L1=Monday 2pm / Friday 9:30am</i> <i>L2=Wednes 3:30pm / Friday 3:30pm</i> <i>L3=Wednes 4:30pm / Friday 4:30pm</i>	<i>Write L1 or L2 or L3:</i>
Your lab: <i>LA1= Monday 3pm</i> <i>LA2= Tuesday 10am</i> <i>LA3= Monday 9:30am</i> <i>LA4= Tuesday 3:30pm</i> <i>LA5= Wednesday 11am</i>	<i>Write LA1 or LA2 or LA3 or LA4 or LA5:</i>

Instructions

- This is an open book, open notes examination
- **No digital devices are permitted**; so no calculators, phones, smart watches, tablets, etc
- You cannot share anything with anyone after the exam begins
- The highest possible mark is 100
- There are 11 questions in total. Some questions have multiple parts.
- Read each question carefully before answering.
- Write your answers clearly in the space provided **in this exam script**
- You need to return this exam script, all pages, for marking
- Be careful to use capital/small letters at the appropriate places
- Assume that the questions use the version of Python used on the course, Python 3.6
- Assume that the questions use the same modules used on the course e.g. the Turtle module

Q1) 6 marks

```
def check(boyfriends_who_know_python):
    if (boyfriends_who_know_python<=2):
        print("Interesting!")
    elif (boyfriends_who_know_python<=4):
        print("Quite crazy!")
    else:
        print("You are really crazy!!")
        print("Because you have so many boyfriends who know python")
        print("I will do some magic and take some away!")
        boyfriends_who_know_python=boyfriends_who_know_python-3

boyfriends_who_know_python=5
check(boyfriends_who_know_python)
print(boyfriends_who_know_python)
```

What number is printed when the program is run?

If you think the program crashes then write ERROR for your answer.

Answer: _____
Write one single number, or write ERROR.

Q2) 6 marks

When the following program is executed, how many times does the word 'hello' get printed?

```
a=0
while a!=1:
    a=a+1
    for b in range(3):
        for c in range(2, 4):
            print("hello")
```

Answer: _____
Write one single integer number.

Q3) 6 marks

How many **squares** do you see after the following program has finished?

```
import turtle
turtle.tracer(False)

for width in range(50, 101, 50):
    for height in range(50, 300, 50):
        turtle.forward(width)
        turtle.left(90)
        turtle.forward(height)
        turtle.left(90)
        turtle.forward(width)
        turtle.left(90)
        turtle.forward(height)
        turtle.left(90)

turtle.tracer(True)
turtle.done()
```

No error occurs when the program is run.

Write down one number which is the number of **squares** (not rectangles) shown on the screen. **The number must include squares of any size.** In other words, include any squares made up of squares in your counting.

For example, if you thought this was the result:
Then your answer would be 5.



Answer: the total number of squares is _____
Write one single integer number which is ≥ 1

Q4) 8 marks

Here is a program.

```
s=["Snacks", "Shock", "Sand", "Snow", "Silliness"]
t=["Trouble", "Typhoons", "Tears", "Tshirts", "Trump"]
s.insert(1, "Stickers")
s.reverse()
s.append("Sausages")
s.remove("Sand")
this_thing=s[0]
s[0]=s[2]
s[2]=this_thing
t.sort()
t[-1]=t[0]
t.reverse()
print("HKUST is the University of", s[-3][:5], "and", t[1])
```

According to the above program,

HKUST is the University of

_____ and _____

Fill in the two missing words. Write clearly. There are 4 marks for each answer.

You can choose whether or not to use speech marks e.g. "Snow". It won't affect the marking.

Q5) 7 marks

A student is building a program to practise programming random numbers. She wants to create a program which can generate the example results shown on the right:

Here is the student's program. The letters *A B C D E* are just used to refer to some lines of the code, they are not part of the code.

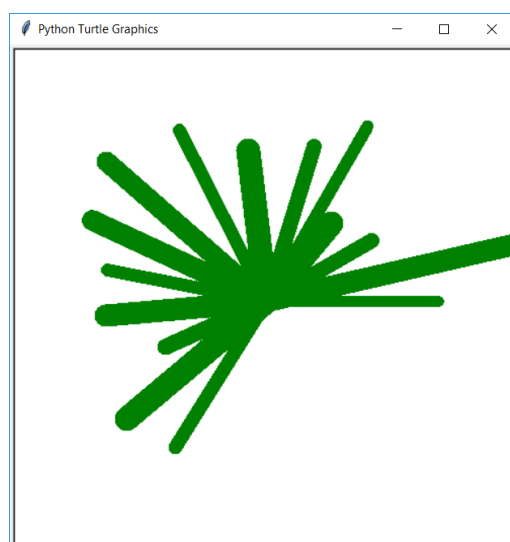
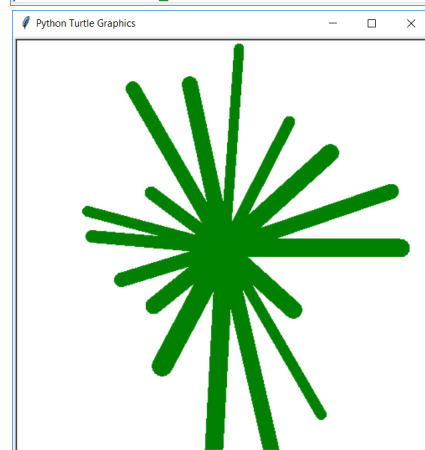
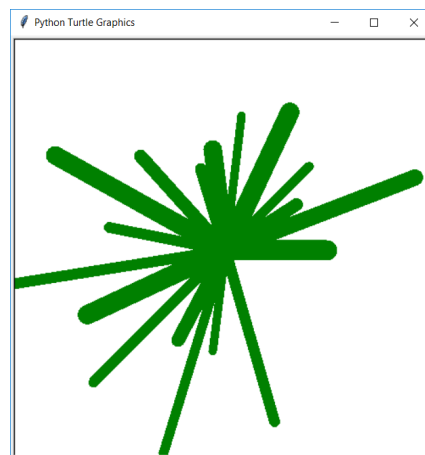
```
import turtle
import random
turtle.setup(500,500)
turtle.width(5)
turtle.color("green")
turtle.speed(10)
A for i in range( random.randint(10,25) ):
B     turtle.width( random.randint(10,25) )
C     turtle.forward( 10 * random.randint(10,25) )
D     turtle.backward( 10 * random.randint(10,25) )
E     turtle.left( random.randint(10,25) )

turtle.hideturtle()
turtle.done()
```

The program code shown above cannot create the three results shown. For the program to be able to correctly generate the results shown, **one line of code needs to be replaced with a new line of code.**

Which line of code needs to be replaced?

You can only choose from the lines labelled A/B/C/D/E. **You don't have to write the new line of code.** Just say which line needs to be replaced.



The line of code which needs to be replaced is (A/B/C/D/E): _____

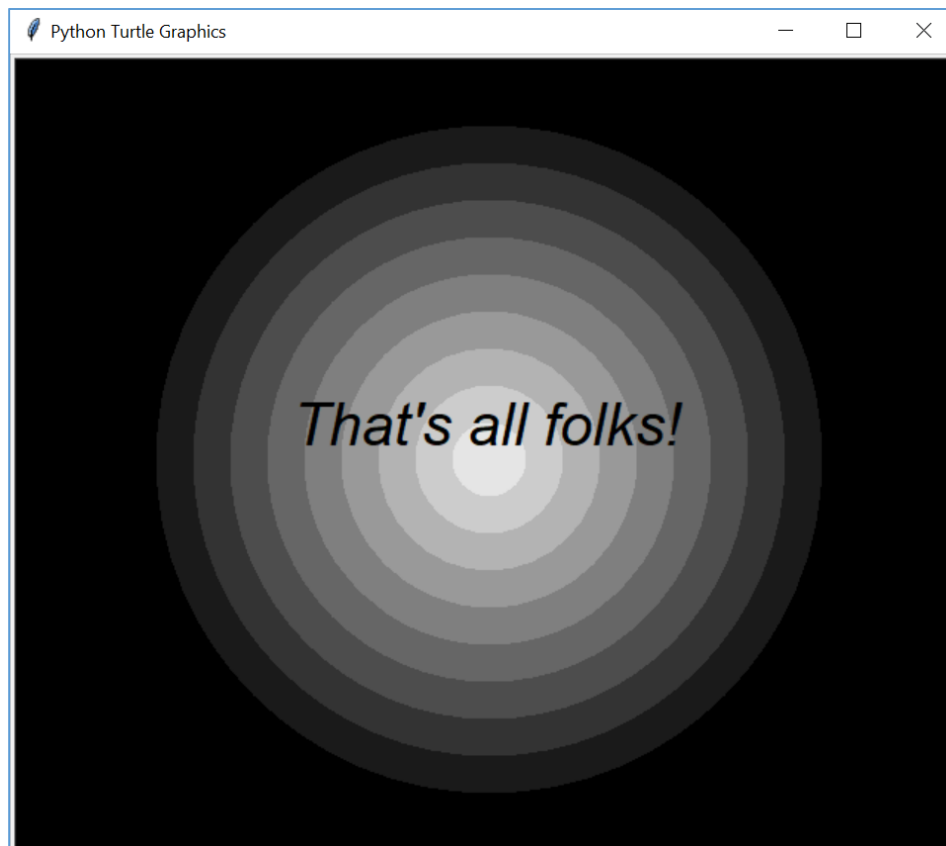
Write A or B or C or D or E.

Q6) 10 marks

Here are some of the colours that can be used in turtle graphics:

grey0	grey1	grey2	grey3	grey4	grey5	grey6	grey7	grey8	grey9
grey10	grey11	grey12	grey13	grey14	grey15	grey16	grey17	grey18	grey19
grey20	grey21	grey22	grey23	grey24	grey25	grey26	grey27	grey28	grey29
grey30	grey31	grey32	grey33	grey34	grey35	grey36	grey37	grey38	grey39
grey40	grey41	grey42	grey43	grey44	grey45	grey46	grey47	grey48	grey49
grey50	grey51	grey52	grey53	grey54	grey55	grey56	grey57	grey58	grey59
grey60	grey61	grey62	grey63	grey64	grey65	grey66	grey67	grey68	grey69
grey70	grey71	grey72	grey73	grey74	grey75	grey76	grey77	grey78	grey79
grey80	grey81	grey82	grey83	grey84	grey85	grey86	grey87	grey88	grey89
grey90	grey91	grey92	grey93	grey94	grey95	grey96	grey97	grey98	grey99

The result of the program shown on the following page is shown here:



```

import turtle
turtle.speed(0)
turtle.bgcolor("black")
turtle.hideturtle()
diameter=500
for x in range(10):
    turtle.color( "grey" + _____ (x*10) )
    turtle.dot(diameter)
    diameter = diameter - _____

turtle.color("black")
turtle.write("That's all folks!", font=("Arial", 30, "italic"), \
            align="center")

turtle.done()

```

You need to write the missing code in the two ____ shown above. You cannot add any other code. You cannot alter any of the code which is already given to you. Some notes:

- For the second answer, you have to appropriately use the minus shown
- Make sure you use speech marks if necessary.

Each part is worth 5 marks.

Q7) 6 marks

```

import turtle

def do_digit1():
    turtle.circle(100)
    turtle.circle(-100)
    turtle.up()
    turtle.forward(150)
    turtle.down()

def do_digit2():
    turtle.circle(100, 180)
    turtle.left(180)
    turtle.circle(-100, 180)
    turtle.left(180)
    turtle.circle(-100, 180)
    turtle.right(180)
    turtle.circle(100, 180)
    turtle.left(180)
    turtle.up()
    turtle.forward(150)
    turtle.down()

```

```
# Main part of the program
turtle.width(5)
turtle.speed(9)
turtle.up()
turtle.goto(-200, 0)
turtle.down()

do_digit1()
do_digit2()
do_digit2()
do_digit2()
do_digit2()
turtle.done()
```

When the program has finished, what number is shown on the screen?

Clearly write one number.

Q8) 15 marks

```
# 012345678901234567890123456789012345678
message="I can't wait for the samsung fold phone"

part1=message[30:]
print(part1)

part2=message[2:5]
print(part2)

part3=message[27:32:3]
print(part3)

part4=message[-3:-20:-10]
print(part4)

part5=message[21:23] + message[-8::-12]
print(part5)
```

It's OK to store an apostrophe (')
in a string " "

What is *part1*? _____

What is *part2*? _____

What is *part3*? _____

What is *part4*? _____

What is *part5*? _____

You can choose whether or not to use speech marks e.g. “for”. It won’t affect the marking.

Each part is worth 3 marks.

Q9) *12 marks* Here is a program.

```
row=2
col=0
locations = [
    ["at HKUST north entrance", "in a steep road", "in the sports center"],
    ["at the red bird", "in Starbucks", "in the library"],
    ["outside the big lecture theatres", "in the Passion restaurant", "in LTG"],
    ["in the south bus area", "in a car park", "in a small road"] ]

while True:
    print("You are", locations[row][col] )
    direction = input("Which way? (n/s/e/w) ")
    if direction == "n":
        if row == 0:
            print("Sorry, you can't go that way!")
        else:
            row = row-1
    elif direction == "s":
        if row == len(locations)-1:
            print("Sorry, you can't go that way!")
        else:
            row = row+1
    elif direction == "e":
        if col == 0:
            print("Sorry, you can't go that way!")
        else:
            col = col-1
    elif direction == "w":
        if col == len(locations[0])-1:
            print("Sorry, you can't go that way!")
        else:
            col = col+1
    if col == 2 and row == len(locations)-1:
        print("You are", locations[row][col] )
        print("You have reached the secret exit!")
        break
```

After you start the program, what text do you need to enter into the program **to get to the secret exit** (which would then trigger the end of the program)? For example, if you think that you need to enter 'n' and then 'e' and then 's' to reach the secret exit, your answer for 'Input 1' would be 'n', and your answer for 'Input 2' would be 'e', and your answer for 'Input 3' would be 's', and you would not write any further answers.

Input 1: _____

Input 2: _____

Input 3: _____

Input 4: _____

Input 5: _____

If you think it is not possible to reach the secret exit when you run the program, write NOT POSSIBLE in your first answer.

If you think there is more than one possible answer then you must provide the shortest path/route that works.

Q10) 8 marks

```
#               Mac, PC, Linux
Computer={  "L1" : [9, 30, 0], \
            "L2" : [20, 32, 0], \
            "L3" : [30, 30, 3]  }

TotalMacs=0
TotalStudents=0

for Section, Information in Computer.items():
    TotalMacs = _____
    TotalStudents = TotalStudents+Information[0]
    TotalStudents = TotalStudents+Information[1]
    TotalStudents = TotalStudents+Information[2]

AverageMac = _____
print("Result:", AverageMac, "% of students who voted use Macs")
```

During lectures earlier in the semester, Prof. Rossiter asked students to raise their hands to indicate if they use a Mac as their main machine, or if they use Microsoft Windows as their main machine, or if they use Linux for their main machine. The results are shown in the dictionary in the above code. (Some students didn't vote for anything, they are ignored in this question).

When the above program is executed, here is the result:

```
Result: 38.311688311688314 % of students who voted use Macs
```

In the two ____ shown above, write the missing code. Write clearly.

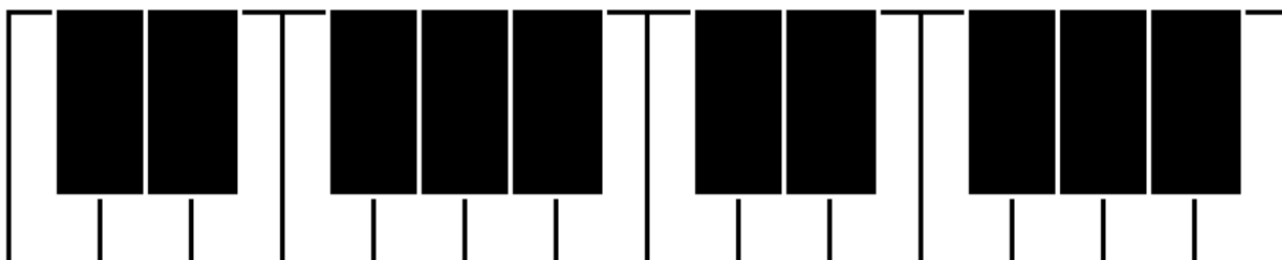
Prof. Rossiter might do the same survey later e.g. a year later. Then the information in the dictionary would be updated e.g. the number of people with Macs or PCs might be different, and there might be more sections. However, all the other code wouldn't be changed. Then the program would be executed to see the result. So that means your code cannot be fixed so it only works for the dictionary data shown above!

Each part is worth 4 marks each.

Q11) 16 marks. Here is a program. The result of the program is shown on the following page.

```
import turtle
allTurtles=[]
startx=-350
# First, let's handle the white keys
x=startx
y=0
xDistanceBetweenKeys=58
WhiteKeyHeightMultiplier=8
KeyWidthMultiplier=3
KeyOutlineWidth=3
for _ in range( _____ ):
    t=turtle.Turtle()
    t.shape("square")
    t.color("black", "white")
    t.shapesize(WhiteKeyHeightMultiplier, \
                 KeyWidthMultiplier, KeyOutlineWidth)
    t.up()
    t.goto(x, y)
    t.down()
    allTurtles.append(t)
    x=x + xDistanceBetweenKeys
# Next, let's handle the black keys
x=startx + (xDistanceBetweenKeys * _____)
y=y + 23
for count in range(13):
    ThisOne=count % _____
    if ThisOne==0 or ThisOne==1 \
        or ThisOne==3 or ThisOne==4 or ThisOne==_____:
        t=turtle.Turtle()
        t.shape("square")
        t.color("white", "black")
        t.shapesize(WhiteKeyHeightMultiplier * .75, \
                     KeyWidthMultiplier, KeyOutlineWidth)
        t.up()
        t.goto(x, y)
        t.down()
        allTurtles.append(t)
        x=x + xDistanceBetweenKeys
turtle.done()
```

The program on the previous page generates this image of a piano keyboard:



In the previous page, you need to write the missing pieces of code in the 4 ____ areas.

There may be more than one way to write an answer which works. However, for each answer, it should be the **simplest (least typing) answer** that works. You cannot change any of the code already given to you. Write clearly.

Each part is worth 4 marks each.

- End of the exam -