THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

COMP1021
Introduction to Computer Science

Final Examination

Tuesday, 29 May 2018

(2 hours duration)

Name:	
Student id:	
Your lecture: ***L1****=Tuesday 4:30pm / Thursday 4:30pm* ***L2****= Tuesday 11am / Thursday 11am* ***L3****= Tuesday 3:30pm / Thursday 3:30pm*	*Write L1 or L2 or L3:*
Your lab: ***LA1****= Friday 2:30pm* ***LA2****= Wednesday 9am* ***LA3****= Wednesday 5pm* ***LA4****= Tuesday 1pm*	*Write LA1 or LA2 or LA3 or LA4:*

Instructions

- This is an open book, open notes examination
- No digital devices are allowed; so no calculators. mobile phones, tablets or computers
- There are 16 questions. Some questions have multiple parts.
- The highest possible mark is 100, the lowest possible mark is zero
- 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

Q1) ***5 marks***

You are working in McDonalds. Your boss uses a Python program to generate the jobs you have to do. The **desired result** of the program is this:

```
['Make french fries', 'Change air filters']
```

Here is the program used by your boss which generates the above result. However, one number is missing in the last line of the code.

```
things_to_do = ['Clean toilets', 'Clean tables', 'Clean chairs', \
  'Make burgers', 'Make chicken nuggets', 'Make french fries', \
  'Change lightbulbs', 'Change air filters', 'Change oil filters', \
  'Make big macs', 'Make cups of tea', 'Make cups of coffee', \
  'Empty bins', 'Clean the big M sign', 'Mop the floor']

print( things_to_do[  5  :  ____________________  :  2  ] )
```

Four of the numbers shown in the answers below can be written in the blank to create the result. **One of the numbers** written below **will not create the desired result**. Choose the answer which **does not create** the desired result.

A) 7
B) 8
C) 9
D) -6
E) -7

Answer (A/B/C/D/E): _____
*Your answer is **one single letter***

Q2) ***5 marks***
What does the following program print?

```
less_mosquitoes=True
nicer_beaches=True
less_salary=True
lower_tax=False

move_to_australia = not less_salary and nicer_beaches or less_mosquitoes \
                  and lower_tax

print(move_to_australia)
```

A) `True`
B) `False`
C) `Yes`
D) `No`
E) `Not`

Answer (A/B/C/D/E): _____
*Your answer is **one single letter***

Q3) ***5 marks***

Here is a Python program.

```
cash=150
coffee_and_cake=40
while cash > coffee_and_cake:
    print("I have ordered some coffee and a cake! ", end="")
    cash = cash - coffee_and_cake
    print("I have drunk/eaten them!")
print("Now I have", cash, "left over")
```

According to the last printed line of the above program, **how much cash** do you have left over?

Answer: _____

Q4) ***8 marks***

Here is a Python program.

```
#       01234567890123456789012345678901234
text = "ryan renalds is the unicorn singer!"
part1 = text[16:19]
part2 = text[28:32] + text[29:32]
part3 = text[13:  :15]
part4 = text[-13:-15:-1] + text[11] + text[8:5:-1]
print(part1)
print(part2)
print(part3)
print(part4)
```

What does *print(part1)* show ? Answer: _____

What does *print(part2)* show ? Answer: _____

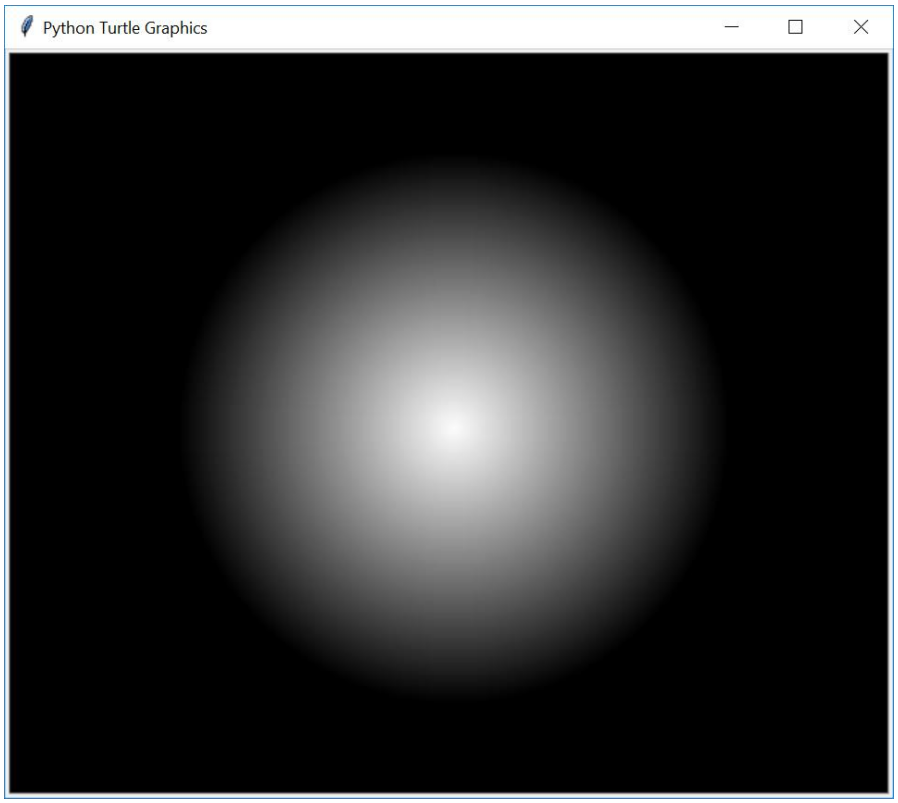
What does *print(part3)* show ? Answer: _____

What does *print(part4)* show ? Answer: _____

2 marks for each totally correct answer.

We will ignore any speech marks when we mark your answer e.g. "cat" 'cat' *and* cat *are the same.*

Q5) ***6 marks***



The program which creates the image shown on the right is given below.

Part of the image is white. Part of the image is black.

As you can see, one line of code is missing.

```
import turtle

turtle.speed(0)

turtle.colormode(255)

turtle.bgcolor("black")

turtle.hideturtle()

diameter=400

for x in range(101):

    The missing line of code goes here

    turtle.dot( max(diameter, 1) )

    diameter = diameter - 4

turtle.done()
```

What is the missing line of code?

A) `turtle.color( x, x, x )`

B) `turtle.color( x*x, x*x, x*x )`

C) `turtle.color( int(x*2.55), int(x*2.55), int(x*2.55) )`

D) `turtle.color( x+255, x+255, x+255 )`

E) `turtle.color( x+100, x+100, x+100 )`

If you think the exact answer is not shown then choose the closest answer.

Answer (A/B/C/D/E): _____

Your answer is ***one single letter***

Q6) *8 marks*

Ms. Chen is thinking of buying a car license plate. She knows that '8' is regarded as a good number in Chinese culture, and '4' is regarded as a bad number. She writes a program to assess a car license plate that she is thinking of buying. To assess a number plate, the programs starts with zero points. Then it adds a point every time it sees an '8' in the car license plate, and it deducts a point whenever it sees a '4' in the car license plate. Finally, it shows the total number of points. Please see the examples below.

1. Example 1 - if the car license plate shown here:

is entered into the program, this is what happens:

```
What license plate do you want to check? 5808
The license plate 5808 has 2 points
(out of a maximum of 4 points)
```

2. Example 2 - if the car license plate shown here:

is entered into the program, this is what happens:

```
What license plate do you want to check? 000088
The license plate 000088 has 2 points
(out of a maximum of 6 points)
```

3. Example 3 - if the car license plate shown here:

is entered into the program, this is what happens:

```
What license plate do you want to check? KF8888
The license plate KF8888 has 4 points
(out of a maximum of 6 points)
```

4. Example 4 - if the car license plate shown here:

is entered into the program, this is what happens:

```
What license plate do you want to check? XVR444
The license plate XVR444 has -3 points
(out of a maximum of 6 points)
```

5. Example 5 - if the car license plate shown here:

is entered into the program, this is what happens:

```
What license plate do you want to check? 48325
The license plate 48325 has 0 points
(out of a maximum of 5 points)
```

The program is shown below. You need to correctly fill in the 4 missing parts. You cannot alter any of the code given to you.

```
car=input("What car license plate do you want to check? ")
points = _____________________________
for letter in _____________________________ :
    if _____________________________ :
        points=points+1
    elif _____________________________ :
        points=points-1
print("The license plate", car, "has", points, "points")
print("(out of a maximum of", len(car), "points)")
```

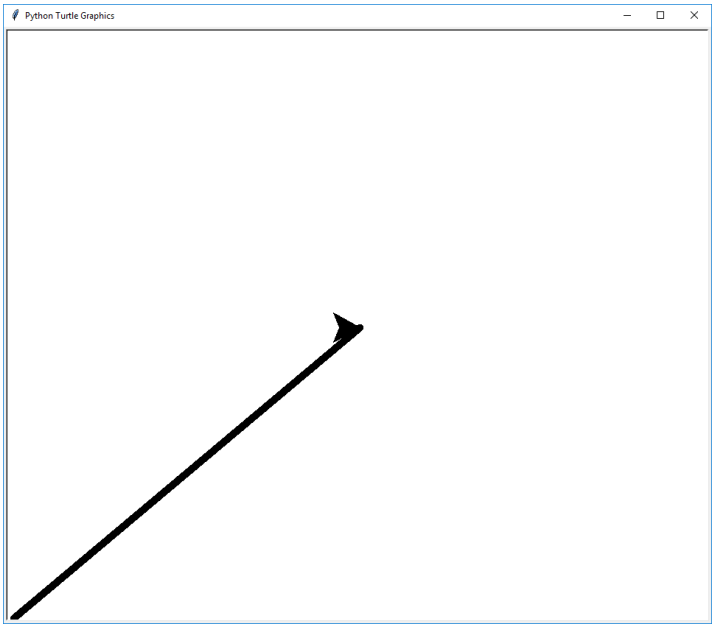
For some of the above blanks there may possibly be more than one answer. In each case you need to enter the simplest answer that works. There are 2 marks for each correct answer.

Q7) ***6 marks***

Here is a program. The comments in the program code help you understand what it does.

```
import turtle
# Here we set the coordinates of the window.
# The four values are: x left, y bottom, x right, y top
turtle.setworldcoordinates(0, 0, 5, 10)
# Now the position (0, 0) is the bottom left corner of the window
turtle.width(10)
turtle.shapesize(4, 4)
turtle.goto(2.5, 5)
turtle.done()
```

The above program creates the following image.



Now here is the main question: a doctor creates a program to show a patient how many red pills, white pills and blue pills a patient must eat each day. The result of the program is shown below.

Python Turtle Graphics

Day	Red	White	Blue
1	0	0	1
2	1	1	2
3	1	1	0
4	2	1	1
5	2	1	2
6	3	1	0
7	3	1	1
8	4	1	2
9	4	1	0
10	5	1	1
11	5	1	2
12	6	1	0
13	6	1	1

Here is the code created by the doctor. You need to identify what the three missing parts are.

```
import turtle
numberOfDays=13
textParams=("Arial", 15, "normal")
turtle.speed(0)
turtle.setworldcoordinates(0, 0, 4, numberOfDays+1)
turtle.up()
turtle.goto(0,numberOfDays)
turtle.write("Day", font=textParams)
turtle.forward(1)
turtle.write("Red", font=textParams)
turtle.forward(1)
turtle.write("White", font=textParams)
turtle.forward(1)
turtle.write("Blue", font=textParams)
for day in range(1, numberOfDays+1):
    turtle.goto(0, numberOfDays-day)
    turtle.write(day, font=textParams)
    turtle.forward(1)
    turtle.write( part 1 , font=textParams)
    turtle.forward(1)
    turtle.write( part 2 , font=textParams)
    turtle.forward(1)
    turtle.write( part 3 , font=textParams)

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

2 marks for each correct answer

For each of ***part 1***, ***part 2***, and ***part 3***, choose one of these:

A) `3-day%3`

B) `int(day/2)`

C) `1%day`

D) `day*7%3`

E) `day**2%3`

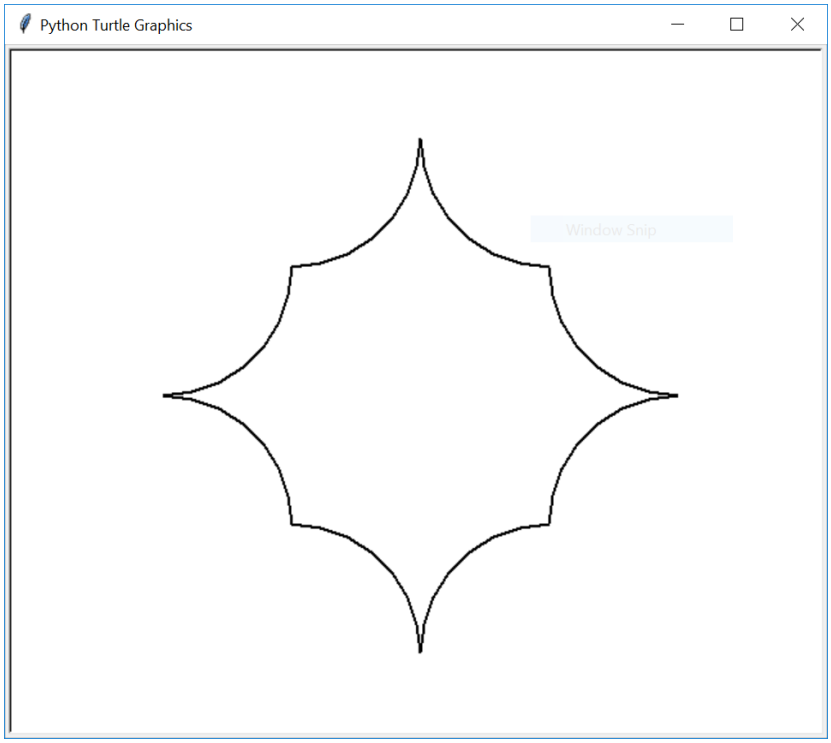
For ***part 1***, your answer is (A/B/C/D/E): _____

For ***part 2***, your answer is (A/B/C/D/E): _____

For ***part 3***, your answer is (A/B/C/D/E): _____

Q8) ***6 marks***

Here is the result of running a Python program.



Which one of the following programs was executed?

A)

```
import turtle
turtle.width(2)
turtle.hideturtle()
turtle.up()
turtle.down()
for thisloopnumber in range(4):
    turtle.circle(100, 90)
    turtle.left(180)
    turtle.circle(100, 90)
    turtle.left(180)
    turtle.right(90)
turtle.done()
```

B)

```
import turtle
turtle.width(2)
turtle.hideturtle()
turtle.up()
turtle.goto(-100, 100)
turtle.down() # The program code for B is continued on the next page
for thisloopnumber in range(4):
```

	``
	    turtle.circle(100, 90)
	    turtle.left(180)
	    turtle.circle(100, 90)
	    turtle.left(180)
	    turtle.left(90)
	turtle.done()
	``

C)	```
	import turtle
	turtle.width(2)
	turtle.hideturtle()
	turtle.up()
	turtle.down()
	for thisloopnumber in range(4):
	    turtle.circle(100, 90)
	    turtle.left(180)
	    turtle.circle(100, 90)
	    turtle.left(180)
	    turtle.right(45)
	turtle.done()
	``

D)	```
	import turtle
	turtle.width(2)
	turtle.hideturtle()
	turtle.up()
	turtle.goto(-100, 100)
	turtle.down()
	for thisloopnumber in range(4):
	    turtle.circle(100, 45)
	    turtle.left(180)
	    turtle.circle(100, 45)
	    turtle.left(180)
	    turtle.left(90)
	turtle.done()
	``

Answer (A/B/C/D): _____

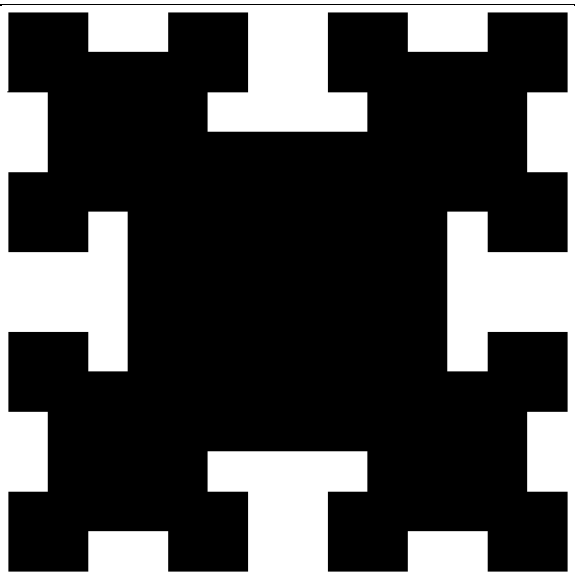
*Your answer is **one single letter***

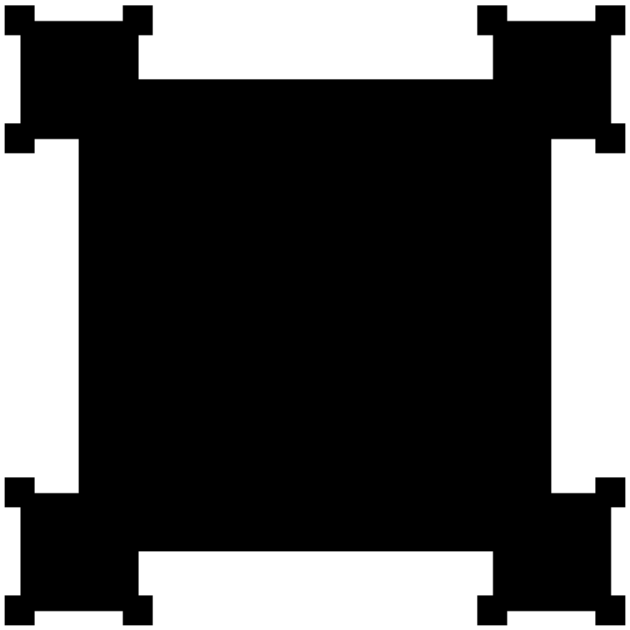
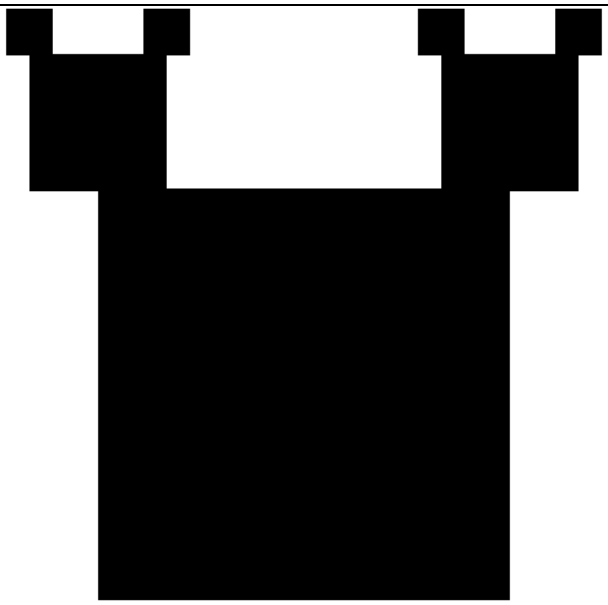
Q9) ***6 marks***

Here is some code.

```
import turtle
max_depth = 2
def do_that_thing(x, y, size, depth):
    turtle.goto(x-size/2, y-size/2)
    turtle.begin_fill()
    for _ in range(4):
        turtle.forward(size)
        turtle.left(90)
    turtle.end_fill()
    if depth < max_depth:
        do_that_thing(x + size /2, y - size / 2, size / 2, depth + 1)
        do_that_thing(x + size /2, y + size / 2, size / 2, depth + 1)
        do_that_thing(x - size /2, y - size / 2, size / 2, depth + 1)
        do_that_thing(x - size /2, y + size / 2, size / 2, depth + 1)
turtle.hideturtle()
turtle.speed(0)
turtle.up()
do_that_thing(0, 0, 300, 0)
turtle.done()
```

What is the result? Select one of the following.

A)	

B)	
C)	
D)	

Answer (A/B/C/D): _____

*Write **one letter** for your answer*

Q10) *6 marks*

The following program plays a famous song called ‘Mary had a little lamb’.

```
1  import time
2  import playsound

3  music = [[0.0, 64], [0.3, 62], [0.6, 60], [0.9, 62], \
3             [1.2, 64], [1.5, 64], [1.8, 64], [2.4, 62], \
3             [2.7, 62], [3.0, 62], [3.6, 64], [3.9, 67], \
3             [4.2, 67], [4.8, 64], [5.1, 62], [5.4, 60], \
3             [5.7, 62], [6.0, 64], [6.3, 64], [6.6, 64], \
3             [6.9, 64], [7.2, 62], [7.5, 62], [7.8, 64], \
3             [8.1, 62], [8.4, 60] ]

4  for i in range(len(music)):
       # Extract the note from the data
5      note_information = music[i]

       # Play the note
6      playsound.play("guitar_sounds/guitar0" + \
6                     str(note_information[1]) + ".wav")
       # Wait for some time
7      if i + 1 < len(music):
8          next_note_information = music[i + 1]
9          time.sleep(next_note_information[0]-note_information[0])
```

The numbers on the left simply indicate the line number. They are not part of the code.

The program works by using lots of different soundfiles, which are stored in a folder inside the folder which contains the Python program. The folder is called *guitar_sounds*. One soundfile contains one note of a guitar. The number of the soundfile indicates the pitch number. For example, if you play guitar060.wav you will hear note 60 of the guitar (a pitch which is roughly in the middle of the range of pitches a guitar can play).

It is possible to amend **one single line of code** to make the music play twice as fast. For this question, consider lines 3, 4, 5, 8, and 9. It is possible to alter four of these lines (just that **one** line of code altered) to make the music play twice as fast. However, one of these lines cannot be altered to make the music play twice as fast.

Which one of lines 3/4/5/8/9 **cannot be altered to make the music play twice as fast**?

Answer: line _____ cannot be altered (by itself) to make the music twice as fast.

Write one single integer number, which is 3 or 4 or 5 or 8 or 9

(Note that you do not have to write down any new lines of code for this question).

Q11) ***6 marks***

An instructor makes a Python program to help provide an indication of how much time has passed during an exam. The idea is that as soon as an exam begins, the instructor starts the program and enters how many hours the exam lasts. No further input is needed. The program then shows a message on the screen and plays an alarm sound every half hour through the exam, and also shows a message on the screen and plays an alarm sound at the end of the exam. Here is an example of the program being used:

```
How many hours long is the exam? 0.5
- the end of exam alarm has been set for 30 minutes later
```

Here is another example:

```
How many hours long is the exam? 1
- a half hour alarm has been set for 30 minutes later
- the end of exam alarm has been set for 60 minutes later
```

Here is another example:

```
How many hours long is the exam? 2
- a half hour alarm has been set for 30 minutes later
- a half hour alarm has been set for 60 minutes later
- a half hour alarm has been set for 90 minutes later
- the end of exam alarm has been set for 120 minutes later
```

Here is another example:

```
How many hours long is the exam? 3.5
- a half hour alarm has been set for 30 minutes later
- a half hour alarm has been set for 60 minutes later
- a half hour alarm has been set for 90 minutes later
- a half hour alarm has been set for 120 minutes later
- a half hour alarm has been set for 150 minutes later
- a half hour alarm has been set for 180 minutes later
- the end of exam alarm has been set for 210 minutes later
```

In the above examples bold is used to show the user's input.

The program code is shown below and on the following page. Make sure you read the comments in the code. The program uses 'playsound.py', which you used in the game lab. Three questions are shown on the following page.

```
import turtle
import playsound

def clearAllText():
    global textTurtle
    textTurtle.clear()
```

You haven't learned .ontimer() in this semester and won't be able to answer this question.

```
def halfHourMessage():
    global textTurtle
    clearAllText()
    textTurtle.write("Half an hour has passed!", font=("Arial",30,"normal"), \
                align="center")
    playsound.play("ding.wav")
    # Clear the half hour message 1 minute later
    turtle.ontimer(clearAllText, [          ])
```

Part 1. Which one of the following goes here?

A) `60*oneSecond`
B) `6*oneSecond`
C) `1000*oneSecond`
D) `oneSecond` Answer (A/B/C/D): ____

```
def endOfExamMessage():
    global textTurtle
    clearAllText()
    textTurtle.write("The exam has finished!", font=("Arial",30,"normal"), \
                align="center")
    playsound.play("ding.wav")

# Main part of the program
oneSecond=1000
textTurtle=turtle.Turtle()
textTurtle.hideturtle()
totalTimeHours=input("How many hours long is the exam? ")
totalTimeMinutes=float(totalTimeHours)*60
totalTimeMinutes=int(totalTimeMinutes)

# Now we set all the half hour alarms
count=1
while count [ ] totalTimeMinutes:
    if count%30 == 0:
        turtle.ontimer(halfHourMessage, count*60*oneSecond)
        print("- a half hour alarm has been set for",count,"minutes later")
    [          ]
```

Part 2. Which one of the following goes here?

A) <
B) <=
C) ==
D) >
E) >= Answer (A/B/C/D/E): ____

Part 3. Which one of the following **cannot** go here?

A) `count+=1`
B) `count=count+1`
C) `count=(count+1)%30` Answer (A/B/C):____

```
# Now set the end-of-exam alarm
turtle.ontimer(endOfExamMessage, totalTimeMinutes*60*oneSecond)
print("- end of exam alarm has been set for",totalTimeMinutes,"minutes later")
turtle.done()
```

Q12) ***8 marks***

This semester Prof. Rossiter was teaching another course in addition to COMP1021. He was teaching COMP4431 Multimedia Computing. In that course the marks were allocated like this:

- Midterm exam: 15%
- Final exam: 40%
- Lab projects: 45%

For this question we are only interested in the lab projects. Here is a slide given to COMP4431 students in the first lecture.

Lab projects (45%)

1. Music processing

2. Audio processing

3. Image processing

4. Video processing

• One of these is worth 0%

– We won't mark one of these four, you choose which one

• Two of these are worth 10% each

– Two of these four will be worth 10%, you choose which two

• One of these is worth 25%

– Choose one of the four areas, finish the work, then add something extra which is not taught in the labs! It will then be worth 25%

Here is an example of an arrangement which students could choose:

Music project=10%, Audio project=10%, Image project=0%, Video project=25%

Here is another example of an arrangement which students could choose:

Music project=25%, Audio project=10%, Image project=10%, Video project=0%

Here is an example of an arrangement which students **can't** choose:

Music project=0%, Audio project=10%, Image project=10%, Video project=0%

Here is another example of an arrangement which students **can't** choose:

Music project=25%, Audio project=10%, Image project=10%, Video project=25%

To show all the possible combinations which the students could choose from, Prof. Rossiter wrote a Python program. The **result** of the program is shown below.

```
Music  Audio Image  Video
0      10     10     25
0      10     25     10
0      25     10     10
10     0      10     25
10     0      25     10
10     10     0      25
10     10     25     0
10     25     0      10
10     25     10     0
25     0      10     10
25     10     0      10
25     10     10     0
```

Here is the code which generated the above result. However, 8 parts are missing.

```
print("Music\tAudio\tImage\tVideo")

for music in ____________________________________________ :

    for audio in ____________________________________________ :

        for image in ____________________________________________ :

            for video in ____________________________________________ :

                thislist=[music, audio, image, video]

                if thislist.count(0)==__________________ and \

                    thislist.count(25)== __________________  and \

                    thislist.count(10)== __________________  and \

                        sum(thislist)== 45 :   # sum() adds up everything

                    text=str(music) + "\t" + str(audio) + "\t" + \

                          str(image) + "\t" + str(video)

                    ____________________________________________
```

Fill in the 8 pieces of missing code. You cannot alter any of the code already given to you. There's 1 mark for each correct answer.

Q13) ***6 marks***

Here is a Python program.

```
row=1
col=1
finished=False
locations = [ ["at the red bird", "in the atrium", "in the G/F restaurant"],
 ["in the lab area", "outside the lecture theatres", "in the coffee shop"],
 ["in the south bus area", "in a car park", "in a small road"],
 ["at HKUST south entrance", "at the roundabout", "in the Business School"] ]

while not finished:
    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 == len(locations[0])-1:
            print("Sorry, you can't go that way!")
        else:
            col = col+1
    elif direction == "w":
        if col == 0:
            print("Sorry, you can't go that way!")
        else:
            col = col-1
    if col == 2 and row == len(locations)-2:
        print("You have found a secret tunnel!")
        col=1
        row=1
```

You start the above program and enter the following sequence.

s *Enter* e *Enter* e *Enter*

Enter simply means you press the Enter key after typing the letter. No spaces are entered.

After you have finished entering the above sequence, what is the **name of the location** that you are in?

Answer: The name of the location is**: _____**

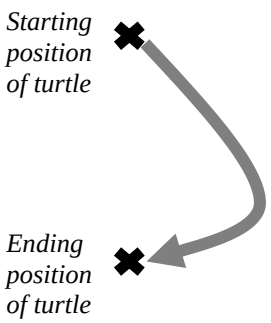
We will ignore any speech marks when we mark your answer e.g. "cat" 'cat' *and* cat *are the same.*

Q14) ***6 marks***

Here is a Python program.

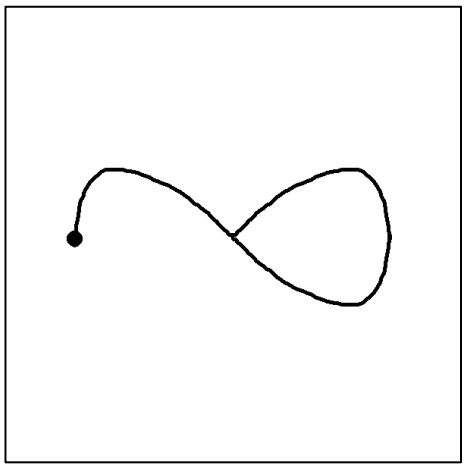
```
import turtle

# Don't show the animation
turtle.tracer(False)

# Create the required turtles

allTurtles = [] # The turtles are put in this list

# The default turtle is the first one in the list
turtle.shape("circle")
turtle.width(5)

allTurtles.append(turtle)

# Add the rest of the turtles in the list
for _ in range(1, 4):
    t = turtle.Turtle()
    t.width(5)
    t.hideturtle()
    allTurtles.append(t)

# Set up the ondrag event
xt = [1, 1, -1, -1]
yt = [1, -1, 1, -1]

def draw(x, y):
    for i in range(4):
        allTurtles[i].goto(x*xt[i], y*yt[i])
    turtle.update() # Update the display

turtle.ondrag(draw)

turtle.update() # Update the display
turtle.done()
```

The user runs the program and drags the main turtle in the manner shown below.

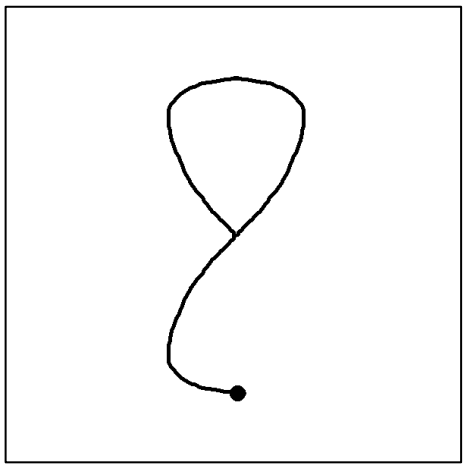


Which one of the following is the resulting display?

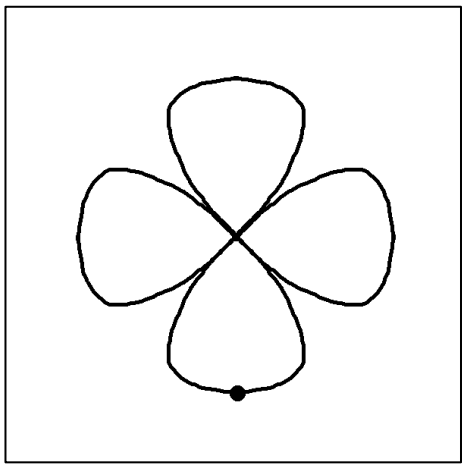
A)



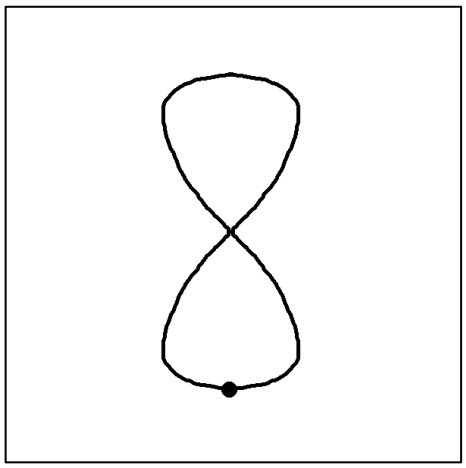
B)



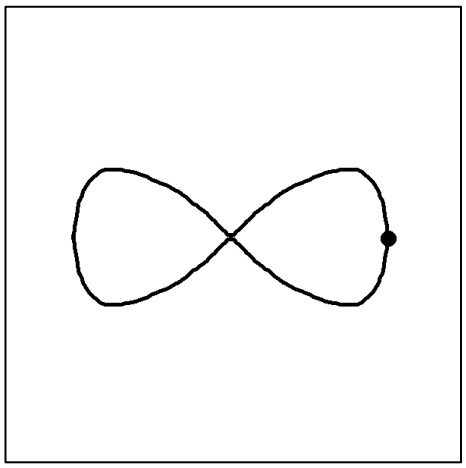
C)



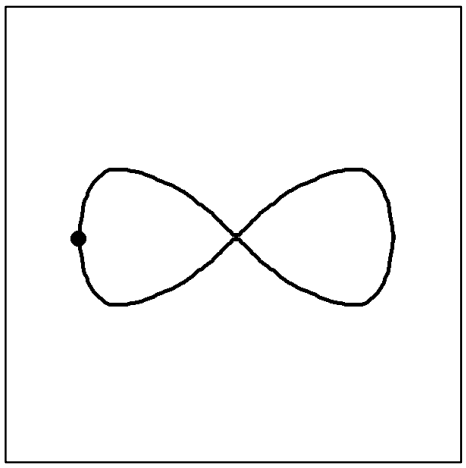
D)



E)



F)



If you cannot see the exact answer then choose the closest answer.

Answer (A/B/C/D/E/F): _____

*Write **one letter** for your answer*

Q15) ***6 marks***

The lyrics for a famous song called 'Do-Re-Mi' have been stored in a text file called `q15_song_lyrics_data.txt`, which is shown below.

```
Let's start at the very beginning
A very good place to start
When you read you begin with ABC
When you sing you begin with Do, Re, Mi, Do, Re, Mi
The first three notes just happen to be
Do, Re, Mi, Do, Re, Mi
Do, Re, Mi, Fa, So, La, Te
Oh let's see if I can make it easier
Doe - A deer, a female deer
Ray - A drop of golden sun
Me - A name I call myself
Far - A long, long way to run
Sew - A needle pulling thread
La - A note to follow sew
Tea - A drink with jam and bread
```

Each of the lines in the text file has a '\n' at the end of it.
The following program is used to process the text file.

```
mydict = {}
myfile = open("q15_song_lyrics_data.txt", 'r')
for line in myfile:
    line = line.rstrip()
    words = line.split(" ")
    if words[1] == "-":
        thisKey=words[0].lower()
        mydict[thisKey] = \
           line[ len(words[0]) + 1 + len(words[1]) + 1 :  ]
myfile.close()
if "tea" in mydict.keys():
    print("The answer is:", mydict["tea"])
else:
    print("It will remain a mystery...")
```

What does the program print when it is executed?

A) `The answer is: A deer, a female deer`

B) `The answer is: A drop of golden sun`

C) `The answer is: A name I call myself`

D) `The answer is: A long, long way to run`

E) `The answer is: A needle pulling thread`

F) `The answer is: A note to follow sew`

G) `The answer is: A drink with jam and bread`

H) `The answer is: Doe`

I) `The answer is: Ray`

J) `The answer is: Me`

K) `The answer is: Far`

L) `The answer is: Sew`

M) `The answer is: La`

N) `The answer is: Tea`

O) `It will remain a mystery...`

P) `What does the fox say?`

Answer (A/B/C/D/E/F/G/H/I/J/K/L/M/N/O/P): _____

*Write **one letter** for your answer*

Q16) ***7 marks***

A few images are given in the same directory as a Python program:

`ustr1c1.gif`

`ustr1c2.gif`

`ustr2c1.gif`

`ustr2c2.gif`

The complete filenames are shown above, underneath the images.

We want to load these image files into the turtle window so that when the user looks at the final result they see this:

The following code loads the image files into the correct position.

```
 1  import turtle

 2  for r in range(1,3):
 3      for c in range(1,3):

 4          x = c * 70
 5          y = -r * 110

 6          newTurtle = turtle.Turtle()

 7          newTurtle.up()

 8          newTurtle.goto(x, y)

 9          thisFilename="image" + r + c + ".gif"

10          turtle.addshape(thisFilename)
11          newTurtle.shape(thisFilename)

12  turtle.done()
```

The numbers on the left simply indicate the line number, they are not part of the code. The above code does not produce the image shown at the top of the page. One of lines 1/6/7/9/10/11 is incorrect.

Part A. Which one of lines 1/6/7/9/10/11 is **incorrect**? **Answer**: Line ___ is incorrect. *3 marks.*

Part B. Write the correct code for that line in the space below. *4 marks.*

You can totally ignore indentation when you write your answer. In other words, don't put any gaps at the start of your answer. Your answer must be **clear and totally correct** to get the marks.

- End of the exam -