

**COMP1021**  
**Introduction to Computer Science**

**Final Examination**

Monday, 27 May 2019

12:30pm – 2:30pm

(2 hours duration)

Your full name, as shown on your HKUST card:	
Your HKUST student ID:	
Your COMP1021 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 COMP1021 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 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
- 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

```
funny=True
```

```
clever=True
```

```
rich=False
```

```
cute=False
```

```
old=False
```

```
has_car=False
```

```
has_house=False
```

```
has_HKID=True
```

```
has_diseases=False
```

```
result= not has_diseases and not old and cute and funny and clever \
        or rich and has_HKID and has_car and has_house
print(result)
```

When it is executed, what does the above program print?

Answer: \_\_\_\_\_

Q2) 5 marks

*Be extra careful with this question!*

```
# 0123456789012345789012345678901234567890
x="Game of thrones has finished! Noooooooooo!"
```

print(x[24:27])                      Answer: \_\_\_\_\_ 1 mark

print(x[-19:-25:-3])                Answer: \_\_\_\_\_ 1 mark

print(x[8:10]+x[3])                 Answer: \_\_\_\_\_ 1 mark

print(x[-20:-29:-3]+x[8])           Answer: \_\_\_\_\_ 1 mark

print(x[11:14])                      Answer: \_\_\_\_\_ 1 mark

Q3) 5 marks

```
import turtle
```

```
turtle.up()
```

```
turtle.goto(-200,200)
```

```
gap=50
```

```
n=8
```

```
special=_____
```

*Don't write a formula, or a fraction, or an expression. Just write one integer number.*



```
for y in range(n):
```

```
    for x in range(n):
```

```
        if x % special == y % special:
```

```
            turtle.down()
```

```
            turtle.dot(gap*.8)
```

```
            turtle.up()
```

```
        turtle.forward(gap)
```

```
    turtle.backward(n*gap)
```

```
    turtle.right(90)
```

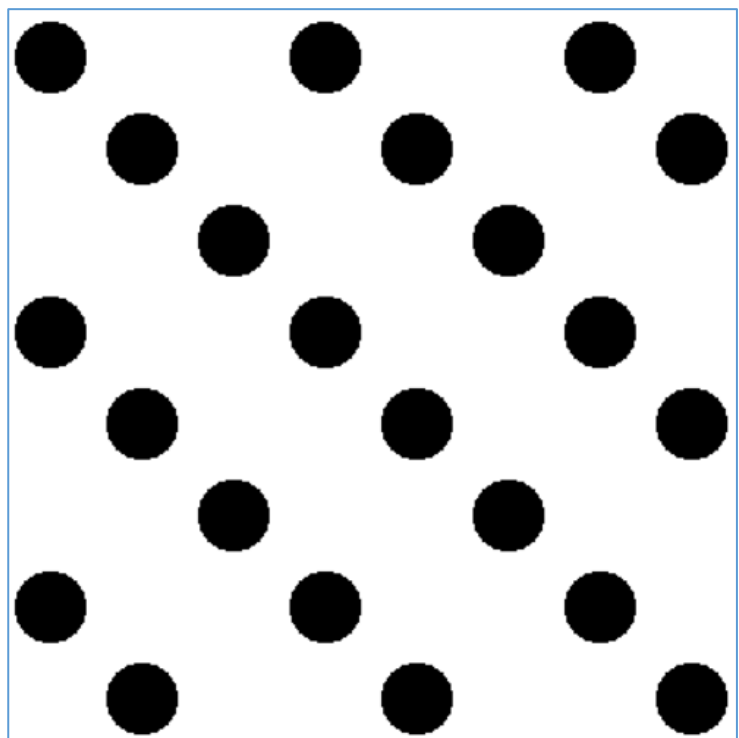
```
    turtle.forward(gap)
```

```
    turtle.left(90)
```

```
turtle.done()
```

The above code produces this result:

In the space shown in the code above, you need to fill in the value of *special*.



**Q4) 5 marks**

```
savings=100000
```

```
cost_of_hk_IT_masters=150000
```

```
cost_of_hk_rent=50000
```

```
cost_of_hk_food=50000
```

```
cost_of_hk_transport=10000
```

```
hk_inflation=1
```

```
hk_inflation/=100
```

```
hk_money_need_to_borrow=savings-cost_of_hk_IT_masters-cost_of_hk_rent- \
    cost_of_hk_food-cost_of_hk_transport
```

```
hk_money_need_to_borrow*=(1+hk_inflation)
```

```
hk_money_need_to_borrow*=-1
```

```
cost_of_uk_hairdressing_qualification=100000
```

```
cost_of_uk_rent=40000
```

```
cost_of_uk_food=60000
```

```
cost_of_uk_transport=20000
```

```
uk_inflation=10
```

```
uk_inflation/=100
```

```
uk_money_need_to_borrow=savings-cost_of_uk_hairdressing_qualification \
    -cost_of_uk_rent- \
    cost_of_uk_food-cost_of_uk_transport
```

```
uk_money_need_to_borrow*=(1+uk_inflation)
```

```
uk_money_need_to_borrow*=-1
```

```
stay_in_hk=hk_money_need_to_borrow<uk_money_need_to_borrow
```

```
print(stay_in_hk)
```

Someone is working out what to do a year later:

- study a Masters' degree in Hong Kong, or:
- get a hairdressing qualification in the UK

That person writes the program shown here to help work out which one to choose.

**Remember: no calculators or any other digital devices are allowed for this exam!**

**When it is executed, what does the above program print?**

**Answer:** \_\_\_\_\_

**Q5) 5 marks**

Here is a dictionary.

```
famous_quotes = {  
    "Trade wars are easy to win." : ["Trump", "Donald"],  
    "I have no special talent. I am only passionately curious." :  
        ["Einstein", "Albert"],  
    "Don't try to be like Jackie. There is only one Jackie. Study computers  
    instead." : ["Chan", "Jackie"]  
}
```

**Part A. 2.5 marks**

Fill in the missing code to produce this result:

```
Trade wars are easy to win.  
I have no special talent. I am only passionately curious.  
Don't try to be like Jackie. There is only one Jackie. Study computers instead.
```

```
for thiskey, thisvalue in famous_quotes.items():  
    print( _____ )
```

**Part B. 2.5 marks**

Fill in the missing code to produce this result:

```
Trump  
Einstein  
Chan
```

```
for thiskey, thisvalue in famous_quotes.items():  
    print( _____ )
```

Here are some notes which apply to both the Part A and Part B questions:

- You cannot alter any of the code already given to you.
- Your answer must use the data from the dictionary.
- There's no use of turtle graphics.

Q6) 5 marks

```
import turtle

turtle.tracer(False)

length=100

def triangle1():

    turtle.forward(length)

    turtle.left(120)

    turtle.forward(length)

    turtle.left(120)

    turtle.forward(length)

    turtle.left(120)

def triangle2():

    turtle.forward(length)

    turtle.right(120)

    turtle.forward(length)

    turtle.right(120)

    turtle.forward(length)

    turtle.right(120)

triangle1()

turtle.left(180)

triangle2()

turtle.right(120)

turtle.forward(length)

turtle.left(120)

triangle2()

turtle.hideturtle()

turtle.tracer(True)

turtle.done()
```

When the program has finished, **how many triangles do you see? Carefully count all the triangles, including triangles of any size and any orientation.**

Answer: \_\_\_\_\_

*Write an integer number  $\geq 1$*

Q7) 5 marks. Here is a program.

```
import turtle
import random

def check():
    thisamount=10
    answer=True
    for turtleNumber in range(1,len(allT)):
        if allT[turtleNumber].distance(allT[0]) > thisamount:
            answer=False
    if answer:
        print("Finished!")
    else:
        print("No...")

allT=[]
numberTurtles=5
turtle.colormode(255)
for i in range(numberTurtles):
    cute=turtle.Turtle()
    cute.shape("circle")
    cute.shapesize((i+1)*2,(i+1)*2)
    cute.speed(0)
    cute.ondrag(cute.goto)
    cute.up()
    thisNumber=int(255/numberTurtles*i)
    cute.color(thisNumber,thisNumber,thisNumber)
    cute.goto( random.randint(-200,200),random.randint(-200,200) )
    allT.append(cute)

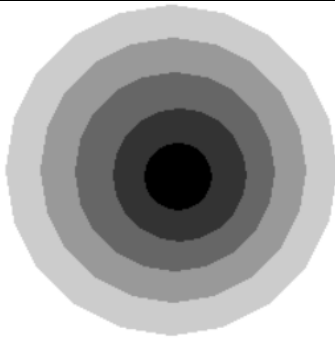
turtle.onkeypress(check, "c")
turtle.listen()
turtle.done()
```

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

When the program is executed the user can drag things. When the user wants to check is he/she has finished, he/she presses the 'c' key. If the user has successfully finished, the message 'Finished!' is shown. If the user has not finished, the message 'No...' is shown.

**One** of the following will produce the 'Finished!' message when the letter *c* is pressed. The other seven will not. Which one will produce the 'Finished!' message when the letter *c* is pressed?

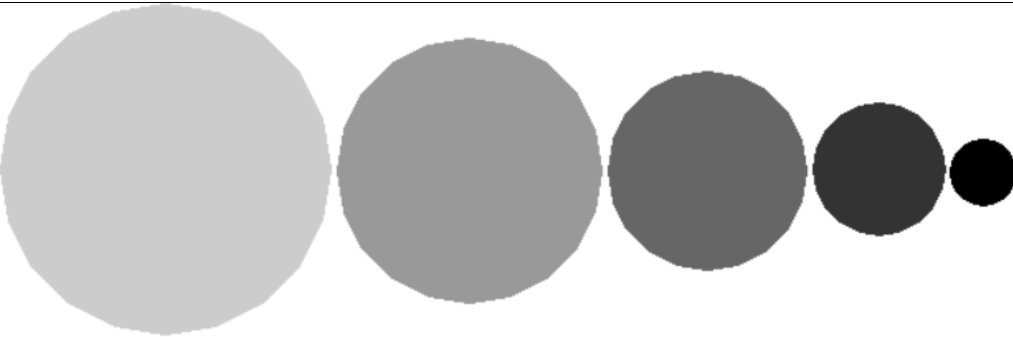
A



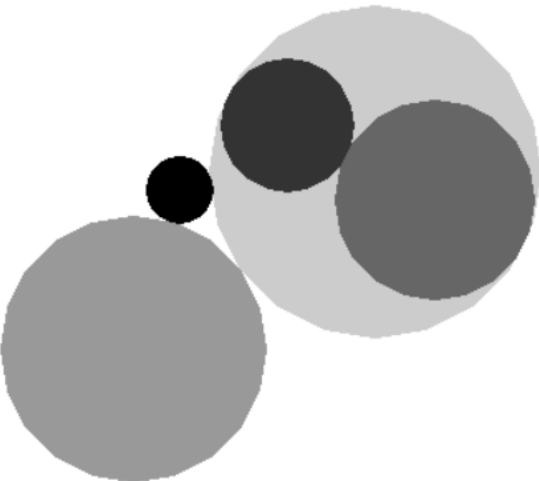
B



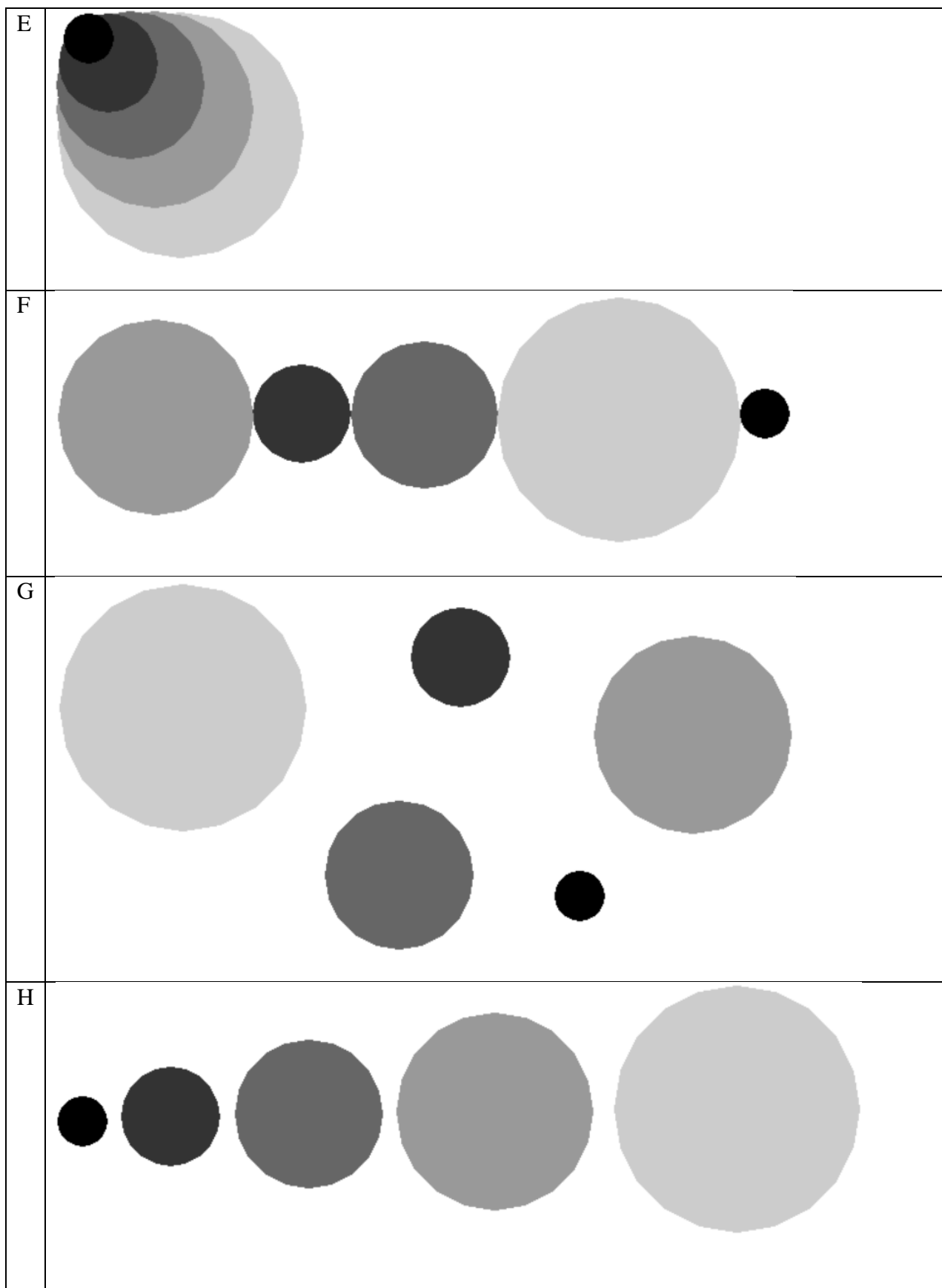
C



D



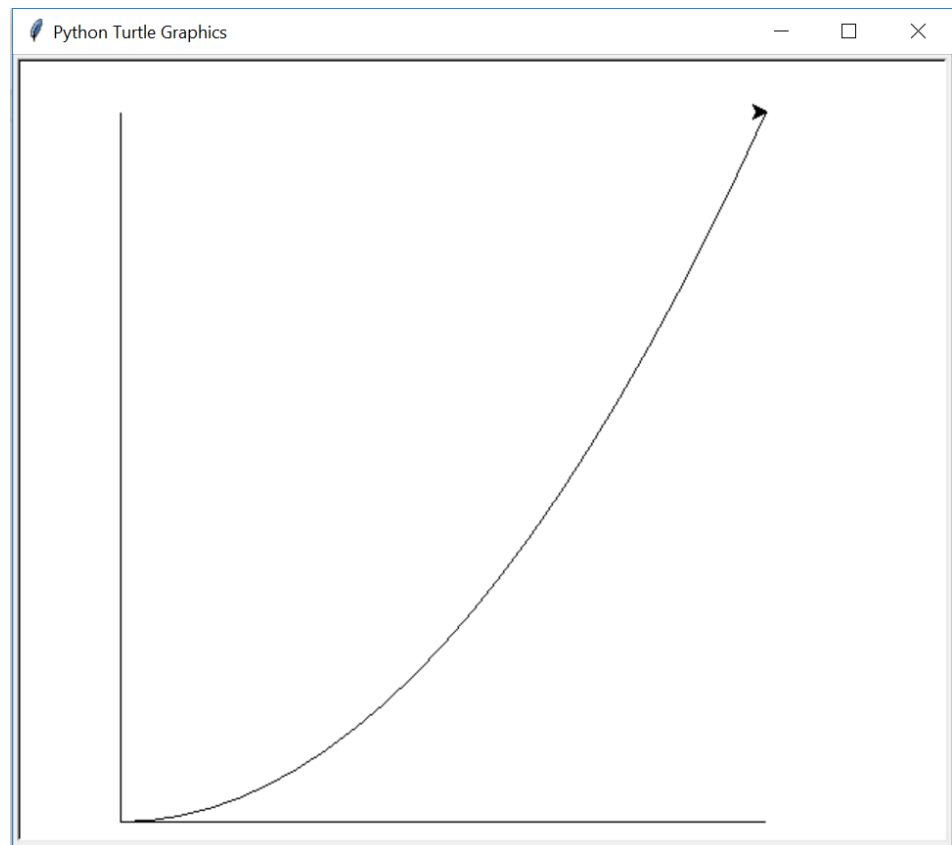




Answer (A/B/C/D/E/F/G/H): \_\_\_\_\_  
*Write one letter*

Q8) 5 marks

Here is the result of the Python program shown below.



```
import turtle
turtle.screensize(500, 500)

turtle.up()
turtle.goto(-250, -250)
turtle.down()
turtle.goto(190,-250)
turtle.up()
turtle.goto(-250, -250)
turtle.down()
turtle.goto(-250, 234)
turtle.up()
turtle.goto(-250, -250)
turtle.down()
for i in range( 1, _____ ) :
    turtle.goto( i*20-250, i*i-250 )
turtle.done()
```

You need to fill in the missing part. For your answer, you need to **write one integer number**. Don't write a formula, or a fraction, or an expression. Just write one integer.

**Q9) 5 marks**

```
def function1():
    print("1")
    function2()
    function2()

def function2():
    print("2")
    function3()

def function3():
    print("3")
    function4()
    function4()
    function4()

def function4():
    print("4")
    function5()

def function5():
    print("5")

function1() # Start here
```

Here is a program. (The 5 functions are shown here next to each other simply to save paper).

A digit is one of 0/1/2/3/4/5/6/7/8/9. When the above program is executed:

**how many times is a digit (any digit) printed?** \_\_\_\_\_

*Write an integer number.*

**Q10) 5 marks**

Someone wants to send a password to his partner in a secret way. First he sends the following Python program to his partner. Later he sends a text message to the partner. The partner runs the program and enters the text message. The secret password is then shown by the program. Here is the program.

```
def create_password(text):
    inc = 1
    value = text[0]
    for pos in range (2, len(text)):
        if text[pos-1] == " ":
            value = value + text[pos+inc]
            inc = inc +1
    return value

plaintext = input("What is the plain text?")
print("The password is", create_password(plaintext))
```

The text message entered into the program is:

unique escapist great fluffy circus appeal

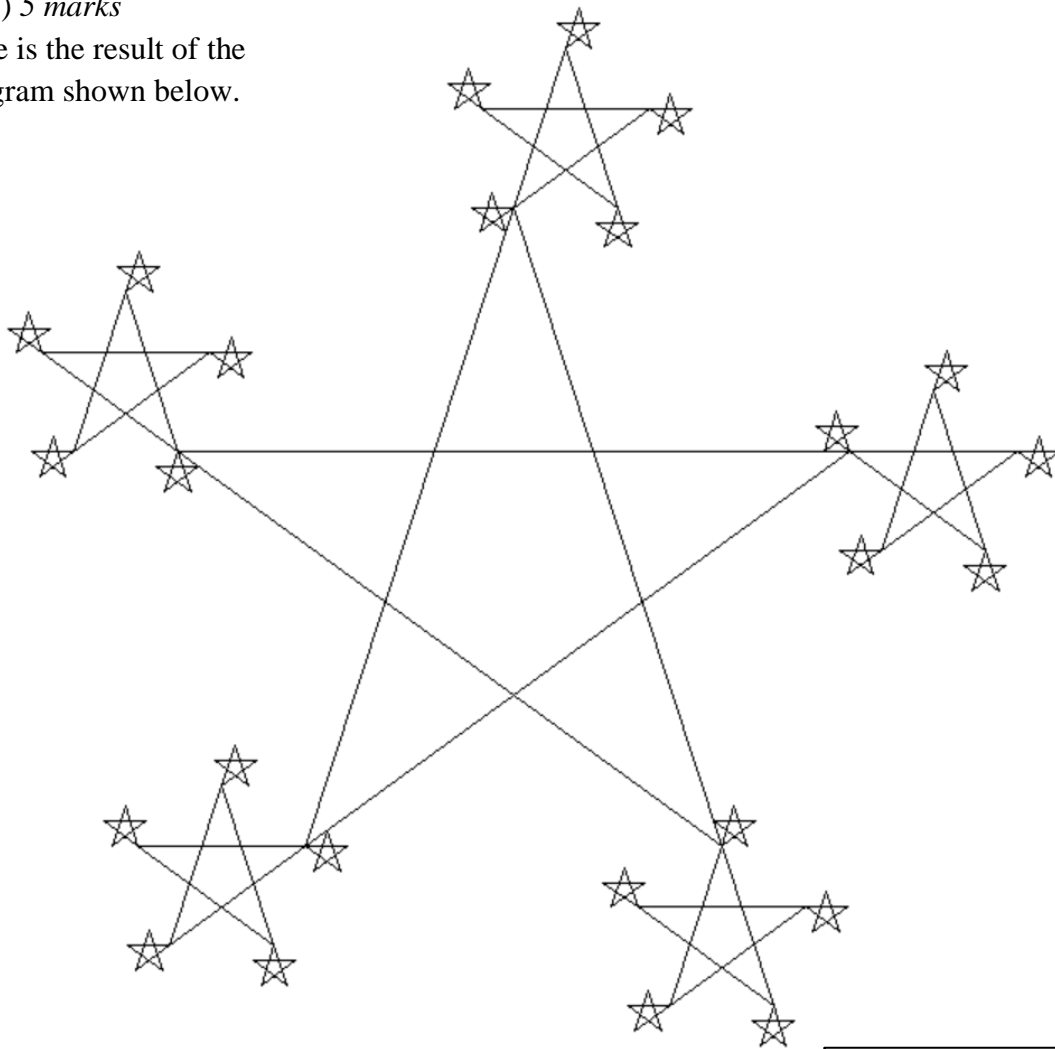
What is the password shown by the program?

Answer: \_\_\_\_\_

*Your answer is one or more letters. Your answer must be completely correct to get the marks.*

Q11) 5 marks

Here is the result of the program shown below.



You need to write the 2 missing pieces. 2.5 marks each.

```
import turtle
```

```
def pretty(quantity, length, angle):
```

```
    if _____ >=25:
```

```
        side = 0
```

```
        while side < quantity:
```

```
            turtle.forward(length)
```

```
            pretty( _____, length/4, angle)
```

```
            turtle.right(angle)
```

```
            side = side + 1
```

```
# The main part

turtle.speed(0)

turtle.up()

turtle.goto(-200, 50)

turtle.down()

pretty(5, 400, 144)

turtle.hideturtle()

turtle.done()
```

Q12) 5 marks

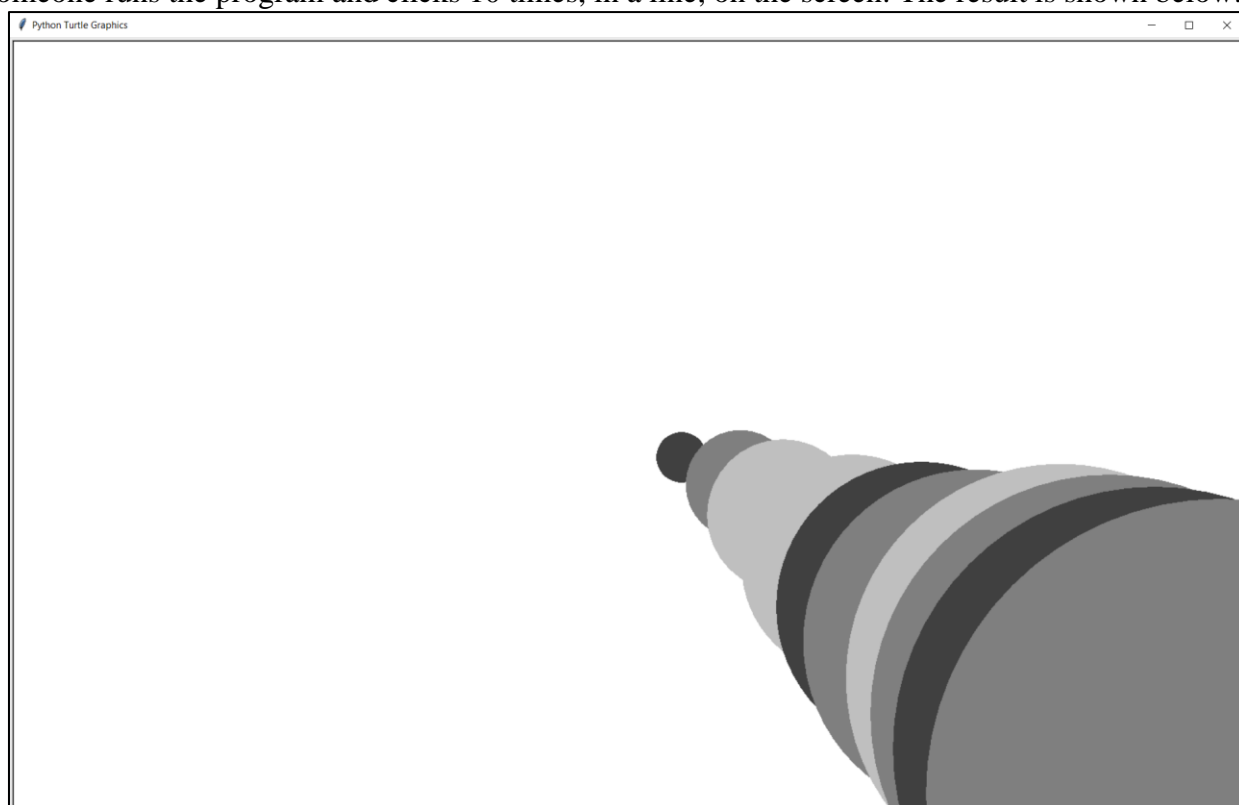
Here is a Python program.

```
import turtle
import random

def show(x, y):
    turtle.goto(x, -y)
    turtle.color(random.choice(["grey25", "grey50", "grey75"]))
    turtle.dot(max(x, 1))

turtle.speed(0)
turtle.shape("circle")
turtle.up()
turtle.onscreenclick(show)
turtle.done()
```

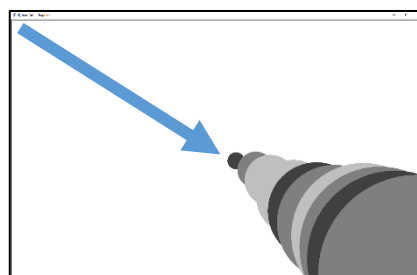
Someone runs the program and clicks 10 times, in a line, on the screen. The result is shown below.



To get the marks for this question **you have to show where the user clicked in a line, and the direction**. You need to do that by **drawing one arrow** on top of the window display shown above.

**For example**, if you thought the display shown above was created when the user clicked in a line from the top left corner of the screen to the middle of the screen you would draw an arrow as shown on the right:

Don't draw lots of arrows. Don't draw lots of circles.  
Just **clearly draw one arrow** in the screen shown above.



Q13) 7 marks

```
import turtle
class Heart:
    def __init__(self, x, y, width, color):
        self.x=x
        self.y=y
        self.width=width
        self.color=color
        self.turtle = turtle.Turtle()
        self.turtle.shape("circle")
        self.turtle.color(self.color)
        self.turtle.up()
        self.turtle.goto(self.x, self.y)
        self.turtle.shapesize(self.width)

    def pump(self):
        self.width=self.width+1
        if self.width>10:
            self.width=1
        self.turtle.shapesize(self.width)
        turtle.ontimer(self.pump, 100)
```

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

# The main part of the program follows.

# Part A. 4 marks. Create a heart object with center (0,0), width 3,  
# using red colour.

myheart = \_\_\_\_\_

# Part B. 3 marks. Make it so that the heart continually pumps, from now  
# onwards until the program stops. Write your answer in the box below.

turtle.done()

You cannot alter any of the code already given to you.

Q14) 9 marks

You decide that from now onwards you will wake up every day at 7am. You write a Python program to help you. Some notes for this question:

- Assume that you run your program at 7am exactly tomorrow, Tuesday 28 May 2019
- 28 May 2019 is day number 148 of the 2019 calendar year
- There are 365 days in the 2019 calendar year
- You will stop the program at the end of the year using *Ctrl c*
- The soundfile library which is used in this code is the same as the library you used on the course e.g. the Fireworks lab. The code for that library is not shown here.

You need to write the missing code in the \_\_\_\_\_ shown below. There are 3 marks for each.

```
import turtle
```

```
import playsound
```

```
def play_alarm_sound():
```

```
    _____ day
```

```
    print("I am about to play the alarm sound")
```

```
    playsound.play("alarm_sound.wav")
```

```
    _____ day
```

```
    if day _____ 365-148:
```

```
        turtle.ontimer(play_alarm_sound, 1000 * 60 * 60 * 24)
```

```
day=0
```

```
play_alarm_sound()
```

```
turtle.done()
```













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

For this answer you need to enter **one** of these:

<    <=    ==    !=    >    >=

Q15) 9 marks

A few images are given in the same directory as the Python program shown lower down the page. Here are the images. As you can see, each image contains a letter. Underneath each image is the name of the image file.

<b>Image:</b>						
<b>Filename:</b>	Image-0-0.gif	Image-0-1.gif	Image-0-2.gif	Image-1-0.gif	Image-1-1.gif	Image-1-2.gif
<b>Image:</b>						
<b>Filename:</b>	Image-2-0.gif	Image-2-1.gif	Image-2-2.gif	Image-3-0.gif	Image-3-1.gif	Image-3-2.gif

All the images are the same size, 80 pixels x 80 pixels.

The following program is executed:

```
1 import turtle
2 x = 100
3 y = 0
4 def createJigsaw():
5     global x, y
6     for a in range(4):
7         for b in range(3):
8             if a+b == 0:
9                 continue
10            elif a+b > 4:
11                continue
12            elif a+b>0:
13                if (a+b)%2:
14                    if a+b != 1:
15                        continue
16                    elif a<b:
17                        continue
18                elif a+b ==2:
19                    continue
20
21 newTurtle = turtle.Turtle()
```

*The program code is continued on the following page*



21	<code>newTurtle.up()</code>
22	<code>newTurtle.speed(0)</code>
23	<code>newTurtle.goto(x, y)</code>
24	<code>x -= 80</code>
25	<code>theFilename="image-" + a + "-" + b + ".gif"</code>
26	<code>turtle.addshape(theFilename)</code>
27	<code>newTurtle.shape(theFilename)</code>
28	<code>newTurtle.ondrag(newTurtle.goto)</code>
29	<code>createJigsaw()</code>
30	<code>turtle.mainloop()</code>

The numbers shown on the left simply indicate the line number, they are not part of the code.

Part A) 3 marks. One line in the `createJigsaw()` function will crash the program (=stop it working). Which line will crash the program? \_\_\_\_\_

*Write the line number*

Part B) 3 marks. Carefully fix the line of code you indicated in part A so that it correctly does what it is trying to do:

---

*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.*

Part C) 3 marks. Assume the program code is correctly fixed. Now, the program is executed. In the space below, show what the program displays:

Q16) 15 marks

You probably already know what a *histogram* is. To help you remember, here is an example of a histogram, from <https://statistics.laerd.com/statistical-guides/understanding-histograms.php>

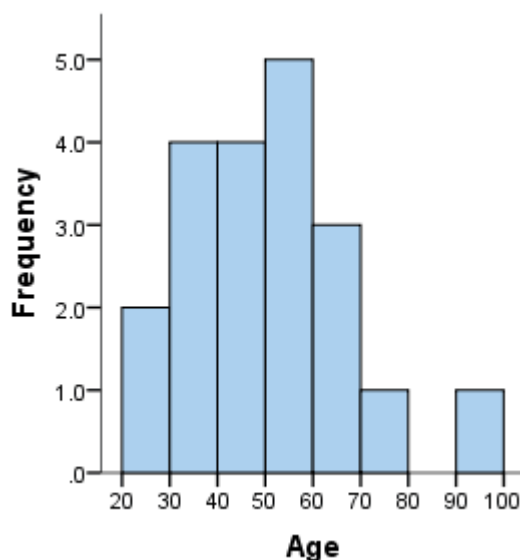
First, take the data:

36    25    38    46    55    68    72    55    36    38  
67    45    22    48    91    46    52    61    58    55

Then put the data into ‘bins’, like this:

Bin	Frequency	Scores Included in Bin
20-30	2	25,22
30-40	4	36,38,36,38
40-50	4	46,45,48,46
50-60	5	55,55,52,58,55
60-70	3	68,67,61
70-80	1	72
80-90	0	-
90-100	1	91

Then you can display the ‘frequency’ number in the bins, like this:



As you can see, the bigger the ‘frequency’ number, the taller the rectangle.

A histogram is useful to show the distribution of data.

Now that you remember what a histogram is, we can start the question on the following page.

Prof. Rossiter wants to build a histogram shows the performance of students taking the COMP1021 midterm exam. First, he makes a file *marks.txt* which contains the name of the student together with their midterm score. When you look at the file using a simple text editor, you see this, below left:

marks - Notepad		
File	Edit	Format View Help
Abra	53	
Absol	53	
Accelgor		88
Aegislash		77
Aerodactyl		94
Aggron	68	
Aipom	24	
Alakazam		0
Alomomola		86
Altaria	44	
Amaura	72	
Ambipom	85	
Amoonguss		79
Ampharos		53

### Some notes

For this question all student names were changed to names of Pokemon, because these are the real marks for the Spring 2019 midterm and the real student names can't be used (the order has also been changed).

All midterm marks are integers.

Pokemon don't have a surname together with other names, they just have one single word for their name.

The entire *marks.txt* file is too long to show here. The entire file has 322 lines. Only the first 14 lines are shown here.

...

Prof. Rossiter writes the code to read the file, create the histogram bins, and then display the resulting histogram. The first bin is for students who got a mark in the range 0 to 9. The second bin is for students who got a mark in the range 10 to 19. The last bin is for students who got full marks (100). This is what happens when the program is executed:

*The program starts*

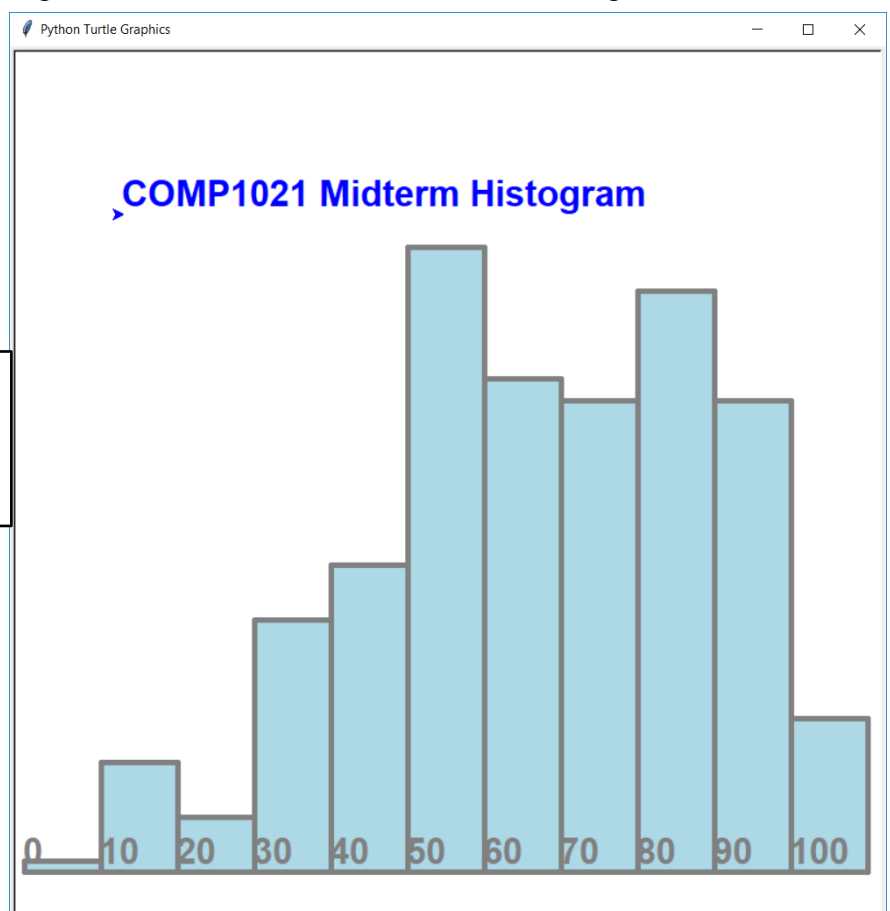


*This text is printed in the shell:*

Here are the histogram values.  
[1, 10, 5, 23, 28, 57, 45, 43, 53, 43, 14]



*This histogram is displayed:*



Here is all the code, which continues on the next 2 pages. You need to fill in the missing pieces of code. You cannot alter any of the code which is given to you. Possibly, there may be more than one answer; marks may be given only to the most straightforward answer. There are 1.5 marks for each correct answer.

```
import turtle
```

```
histogram=[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] # We will use 11 bins in total
```

Write the missing variable name.

```
midterm_file = open("marks.txt", " _____ ")
```

```
for _____ in midterm_file:
```

Write the correct letter here. Speech marks are already written for you, so don't write them again.

```
one_line = one_line.rstrip() # Remove the '\n' on the right
```

```
columns = one_line.split( _____ )
```

Write the correct content here. If your answer needs speech marks then include them. Write clearly.

```
this_name = columns[0]
```

Write the missing part. Read 'Some notes' on the last page first.

```
this_mark = _____
```

Write the missing part. This answer must be an integer.

```
bin_number_to_increase = this_mark // _____
```

```
histogram[bin_number_to_increase]=histogram[bin_number_to_increase]+ \
```

Write the missing part.

```
print() # Print an empty line
```

```
print("Here are the histogram values.")
```

```
print(histogram)
```

*The Q16 program code is continued on the next page*

```
# Let's create a useful function we will use later
```

```
def draw_rectangle( width, height ):
```

```
    turtle.begin_fill()
```

```
    turtle.forward(width)
```

```
    turtle.left(90)
```

```
    turtle.forward(height)
```

```
    turtle.left(90)
```

```
    turtle.forward( _____ )
```



Write the correct variable name here.

```
    turtle.left(90)
```

```
    turtle.forward( _____ )
```



Write the correct variable name here.

```
    turtle.left(90)
```

```
    turtle.end_fill()
```

***The Q16 program code is continued on the next page***

```
# Now we display the histogram
```

```
turtle.width(5)
```

```
turtle.speed(0)
```

```
turtle.color("grey", "lightblue")
```

```
turtle.setup(800, 800)
```

```
turtle.up()
```

```
turtle.goto(-390,-350)
```

```
turtle.down()
```

```
for this_bin_number in _____ :
```

```
    draw_rectangle( 70, histogram[this_bin_number]*10 )
```

```
    turtle.write( str(this_bin_number*10), \
```

```
                  font=("Arial", 25, "bold") )
```

```
    turtle.up()
```

```
    turtle.forward( _____ )
```

```
    turtle.down()
```

```
turtle.up()
```

```
turtle.goto(-300,250)
```

```
turtle.down()
```

```
turtle.color("blue")
```

```
turtle.write("COMP1021 Midterm Histogram", font=("Arial", 25, "bold") )
```

```
turtle.done()
```

Write the missing part. This answer must refer to the list containing the bin data.



Write the missing part.



***This is the end of the Q16 program code - End of the exam -***