

COMP1021
Introduction to Computer Science

Midterm Examination

Monday, 26 March 2018
7pm – 8pm
(1 hour duration)

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

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

Instructions

- This is an open book, open notes examination
- Simple calculators are permitted; smart phones/watches, tablets and computers are not
- You cannot share anything with anyone after the exam begins
- The highest possible mark is 100
- There are 10 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) 7 marks

```
import turtle
turtle.width(5)
for _ in range(10, 40, 10):
    sides=0
    while sides<5:
        turtle.forward(50)
        turtle.right(90)
        turtle.forward(50)
        sides=sides+1
turtle.done()
```

When the loops have all finished, how many squares do you see on the screen?

Answer: _____
Write one single integer number. Your answer is > 0.

Q2) 7 marks

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

girlfriends_quantity=5
check(girlfriends_quantity)
print(girlfriends_quantity)
```

What number is printed when the program is run?

Answer: _____
Write one single number.

Q3) 7 marks

What do you see after the following program has finished?

```
import turtle

for j in range(6):
    for i in range(6):
        turtle.forward(100)
        turtle.left(60)
    turtle.left(60)

turtle.done()
```

- A) A triangle
- B) A square
- C) A smiling face
- D) A car
- E) A chessboard
- F) Something which is not mentioned above

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

Q4) 10 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[ -1 ], "and", t[ -1 ])
```

According to the above program, HKUST is the University of

_____ and _____

*Fill in the two missing words. Write clearly. There are 5 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

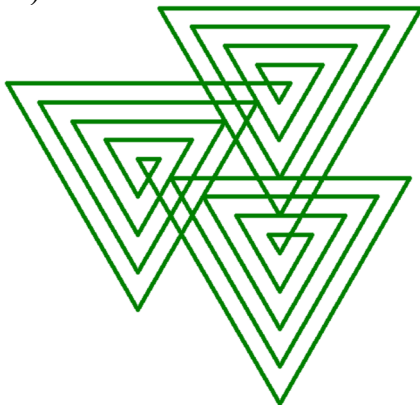
What is the visual output after the following program has finished?

```
import turtle
turtle.width(5)
turtle.color("green")
turtle.speed(10)
turtle.up()
turtle.goto(-100, 50)
turtle.down()

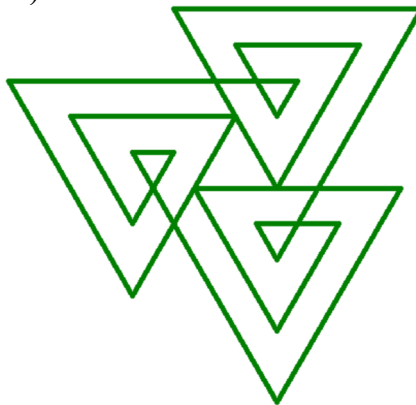
for i in range(3):
    for j in range(7):
        turtle.forward((j + 1) * 40)
        turtle.right(120)

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

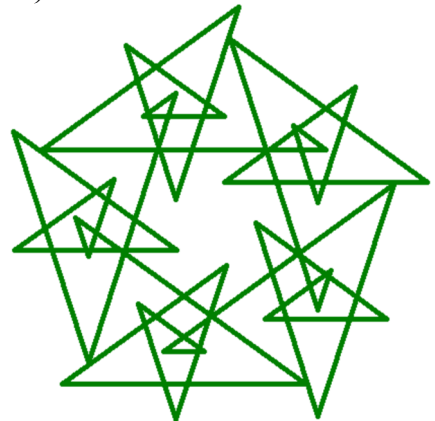
A)



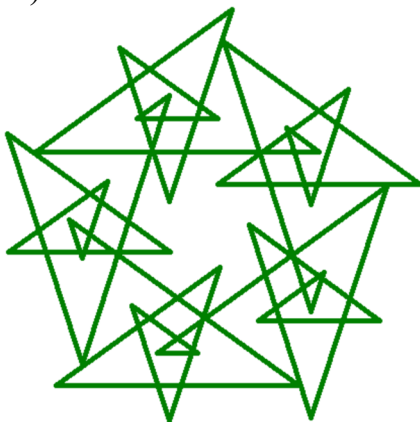
B)



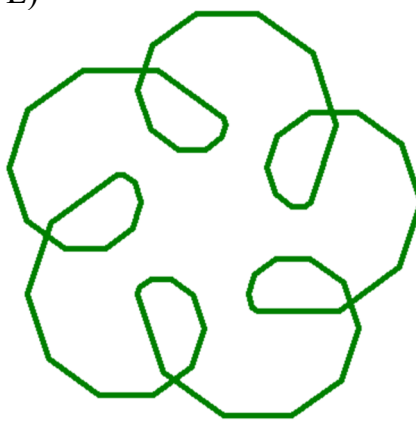
C)



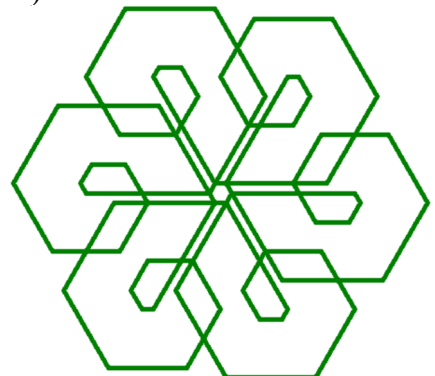
D)



E)



F)



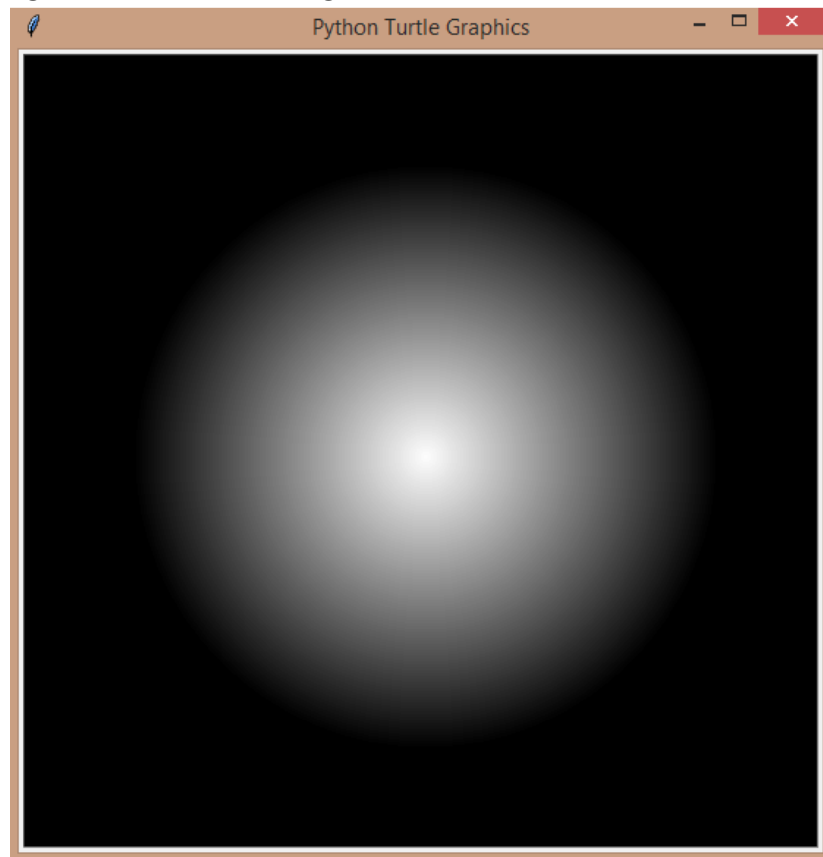
Answer (A/B/C/D/E/F): _____

Q6) 15 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 following program creates this image:



```

import turtle
turtle.speed(0)
turtle.bgcolor( _____ )
turtle.hideturtle()

diameter=400
for x in range(100):
    turtle.color( _____ )

    turtle.dot(diameter)

    diameter= _____ - _____

turtle.done()

```

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

- You have to appropriately use the minus shown in the third answer.
- There may be more than one answer which works. However, the code you write must be the least typing code that works.
- Make sure you use speech marks if necessary.

Q7) 7 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()
turtle.done()
```

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

Clearly write one integer number.

Q8) 16 marks

```
#          012345678901234567890123
message="i have no lemon for tea!"

part1=message[ -3: -1 ] + message[20]
print(part1)

part2=message[ 12 : 9 : -1 ] + message[ 8 : 6 : -1 ]
print(part2)

part3=message[ -2 : -6 : -2 ]
print(part3)

part4=message[ -22 ] + message[ 13 : 10 : -1 ]
print(part4)
```

What is *part1*? _____

What is *part2*? _____

What is *part3*? _____

What is *part4*? _____

You can choose whether or not to use speech marks e.g. “cat”. It won’t affect the marking.
Each part is worth 4 marks each.

Q9) 12 marks Here is a 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)-1:
        print("You are", locations[row][col] )
        print("You have reached the secret exit!")
        finished=True
```

After you start the program, what text do you need to enter into the program in order **to get to the secret exit** (which would then trigger the end of the program)? For example, if you think that if 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 for your first answer.

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

Q10) 12 marks

The following program was sent to you, attached to an email, on Wednesday 7 March 2018.

```
import random
print("Dave's schedule")
print()
maximumCough=48
for day in range(1, 3):
    for hour in range(1, 25):
        print("Day", day, "hour", hour, end=" : ")
        if hour % (day*2) == 0:
            print("sneeze", end=" ")
        for cough in range(random.randint(0, maximumCough)):
            print("cof", end=" ") # 'cough' takes up too much space in
                                # the output, let's use 'cof' instead
        maximumCough=maximumCough-1
        print()
    print()
```

For time management, it may be wise to do this question after you have done the others

Here is some output.

Dave's schedule

Day 1 hour 1 : cof cof cof cof cof cof cof cof cof cof cof cof
Day 1 hour 2 : sneeze cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
Day 1 hour 3 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
cof
Day 1 hour 4 : sneeze cof
Day 1 hour 5 :
Day 1 hour 6 : sneeze cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
Day 1 hour 7 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
Day 1 hour 8 : sneeze cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
cof cof cof cof
Day 1 hour 9 : cof
Day 1 hour 10 : sneeze cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
Day 1 hour 11 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
Day 1 hour 12 : sneeze cof cof cof cof
Day 1 hour 13 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
cof
Day 1 hour 14 : sneeze cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
cof cof cof cof cof cof cof cof cof cof cof cof
Day 1 hour 15 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
cof cof cof cof
Day 1 hour 16 : sneeze cof

Day 1 hour 17 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
 Day 1 hour 18 : sneeze cof cof cof cof cof cof cof cof cof cof cof cof
 Day 1 hour 19 : cof cof cof cof cof cof cof cof
 Day 1 hour 20 : sneeze cof cof cof cof cof cof
 Day 1 hour 21 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
 Day 1 hour 22 : sneeze cof cof cof cof cof
 Day 1 hour 23 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
 Day 1 hour 24 : sneeze cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
 cof cof cof cof cof

Day 2 hour 1 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
 Day 2 hour 2 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
 Day 2 hour 3 : cof cof cof cof cof cof cof
 Day 2 hour 4 : sneeze cof cof cof cof cof cof cof
 Day 2 hour 5 : cof cof cof cof cof cof
 Day 2 hour 6 : cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof cof
 Day 2 hour 7 : cof cof cof cof
 Day 2 hour 8 : sneeze cof cof cof
 Day 2 hour 9 : cof cof cof cof cof cof
 Day 2 hour 10 : cof cof cof cof cof
 Day 2 hour 11 : cof cof cof cof cof
 Day 2 hour 12 : sneeze cof cof
 Day 2 hour 13 : cof cof cof cof cof cof cof cof cof cof cof cof cof
 Day 2 hour 14 : sneeze cof cof cof
 Day 2 hour 15 : cof cof cof cof cof cof cof cof cof
 Day 2 hour 16 : sneeze cof
 Day 2 hour 17 : cof cof cof cof cof
 Day 2 hour 18 :
 Day 2 hour 19 : cof
 Day 2 hour 20 : sneeze cof cof cof cof
 Day 2 hour 21 :
 Day 2 hour 22 : cof cof cof cof
 Day 2 hour 23 : cof cof
 Day 2 hour 24 : sneeze

By looking at the code together with the output, you can find TWO lines of output which **cannot have been generated** by the program. Which two lines of output **cannot have been generated** by the program?

Each part is worth 6 marks. The order of your two answers is not important.

Answer 1: Day _____ hour _____ cannot have been generated.

Answer 2: Day _____ hour _____ cannot have been generated.

- End of the exam -