

Sara Vonk - Marks

Q1 a) Algorithm for tea: 16 /17

b) Different drink selections - 0 /5

c) Storing data on a computer - 1 /3 **Total for Question 1 - 17 /25**

Q2 Flowchart 18/25 **Total for Question 2 - 18/25**

Q3 Python code for 2) above **Total for Question 3 - 25/25**

Q4 a) - An oval in turtle - 9/9

b) A 5 point star in turtle - 8/8

c) A spiral in turtle - 8/8 **Total for Question 4 - 25/25**

OVERALL TOTAL - 85%

Mark signed

85%

Mark agreed

Computational Thinking Test Oct 2017

Name Sara Vank

Student number 1704018

Programme ~~Python~~ Electrical Electronics HND

Answer only the questions you can – if you cannot answer a question, move on to the next question. This test is not negatively marked, so you may wish to guess if unsure. Questions 1 and 2 to be answered on this paper, questions 3 and 4 to be answered using Python and saved (then uploaded via Moodle for staff).

- 1) You have been selected to program a new robot intended to create hot drinks. The robot is capable of following simple, tea and coffee-oriented commands precisely, but has no understanding either of the process, or the fundamental principles which underpin it (e.g. that a kettle requires power). The robot has access to the following items:
- Kettle (initially unplugged)
 - Tea bags
 - Jar of ground instant coffee
 - 1L carton of milk
 - Unopened bag of sugar
 - 1 metal tea spoon
 - 1 large mug
 - Access to a sink for water and an electrical socket for power
- a) You are required to give the robot the set of instructions (algorithm) necessary to successfully make a cup of milky coffee with 1 teaspoon of sugar. Each instruction should be on a new line. Ensure that your instructions are in a logical order and no steps are missed; while highly capable of following instructions, the robot cannot solve problems independently.

(17)

- ① Make sure Kettle is plugged in.
 - ② Fill kettle with water.
 - ③ Place kettle ~~on~~ back on to power stand, push button on side of kettle till light comes on (do not touch kettle till light is off).
 - ④ Find: mug, instant coffee, 1 metal tea spoon and sugar.
 - ⑤ Open lid of instant coffee.
 - ⑥ Put spoon in to instant coffee and pull it out so coffee is flat and level to rim of spoon. ✓ 3
 - ⑦ Put ~~so~~ spoon with coffee in mug.
 - ⑧ Open bag of sugar. ✓ 1
 - ⑨ Put spoon in ~~of~~ and take it out with ~~3~~ sugar flat and level to rim (then put in mug). ✓ 3
 - ⑩ When kettle light is off, pour water in ~~to~~ 2 mug leaving 1 inch gap between liquid in mug and rim of mug. ✓ 3
- (PTO for more space)

- ⑪ Find carton of milk.
- ⑫ open carton of milk.
- ⑬ pour milk in to mug fill ^{✓2} liquid in mug is 3mm from rim of mug.
- ⑭ turn power of kettle off.

Coffee made

No stir

+2

16

- b) The robot is, of course, capable of making many variations of hot drink. The user must be permitted to give information about their drink preferences to the robot before it is created.

0

(5)

- c) If the instructions were converted to code, explain briefly how this data may be stored.
What is the name given to a piece of data stored by a computer?

Code/memory ✓①
Code is ^{converted} ~~converted~~ in to an old code language
which is then converted in to binary (10100110)

(3)

2)

We have been given the requirements for a program which are:

Requirement 1 - Says "Hello" to the user at the start.

Requirement 2- Reads a series of numbers from the user, adding them together.

Requirement 3 - Stops when the user inputs -1 as a value.

Requirement 4 - Once the user elects to stop inputting numbers, outputs the sum of the numbers input.

Requirement 5 - Says "Goodbye" to the user.

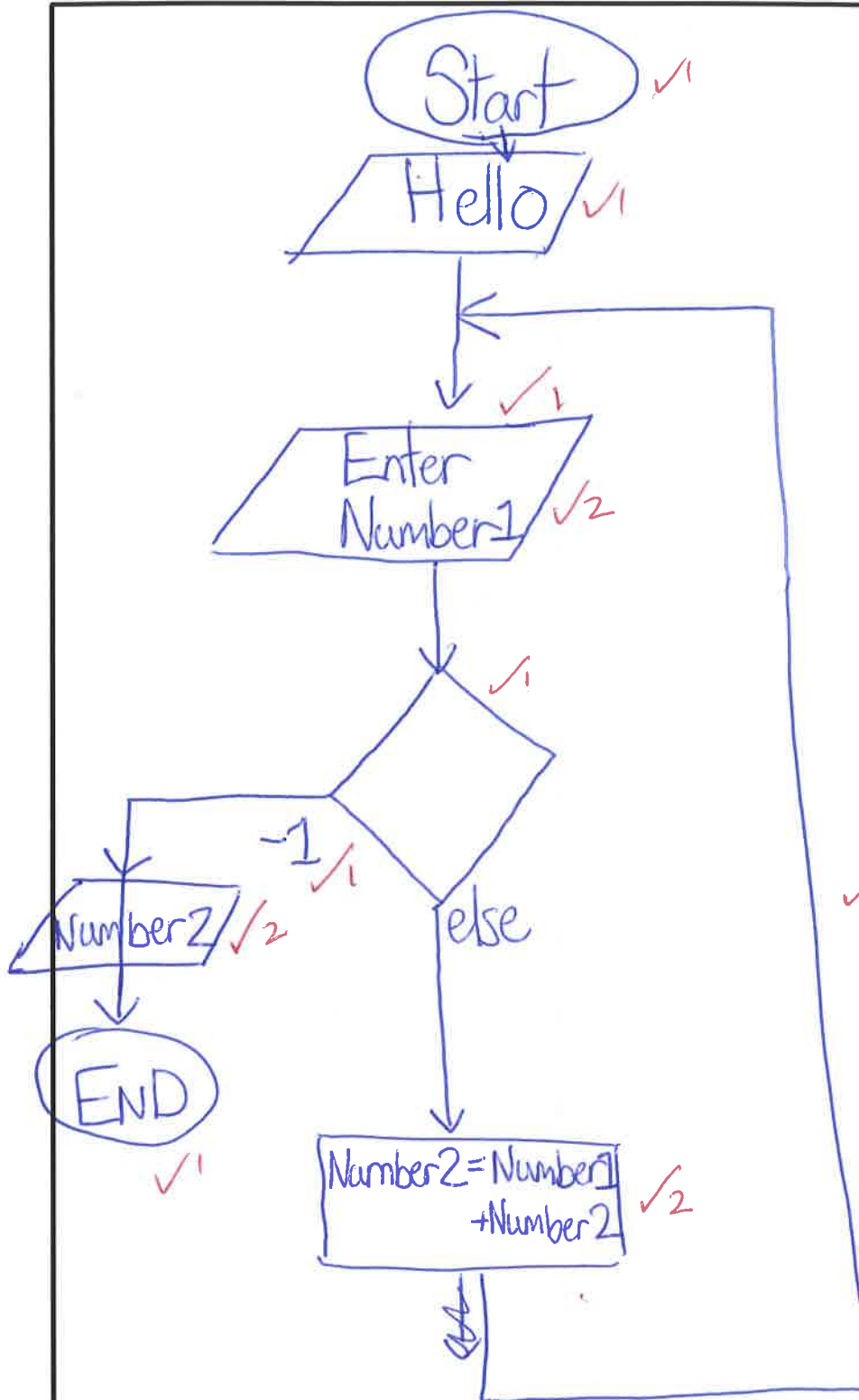
For example, the program may run something like:

```
Hello
Enter number:
8
Enter number:
10
Enter number:
3
Enter number:
-1
Your total is 21.
Goodbye.
```

Of course, the user may enter any set of numbers and get a valid result, here's another example of the program running:

```
Hello
Enter number:
9
Enter number:
-1
Your total is 9.
Goodbye.
```

Write the flowchart for this program:



+2 Logic
+ 2 Boxes
+ 1 Branch.

18

- 3) Now write the in Python code for the program given in 2) above. Remember, the output might look similar to:

```
Hello.  
Enter number (-1 to exit): 8  
Enter number (-1 to exit): 10  
Enter number (-1 to exit): 3  
Enter number (-1 to exit): -1  
Your total is 21  
Goodbye.
```

(15)

- (Advanced) The program should additionally prevent the user from adding negative numbers (other than -1 to exit). For example:

```
Hello.  
Enter number (-1 to exit): 5  
Enter number (-1 to exit): 10  
Enter number (-1 to exit): -5  
You cannot input a negative number.  
Enter number (-1 to exit): 7  
Enter number (-1 to exit): 2  
Enter number (-1 to exit): 6  
Enter number (-1 to exit): -1  
Your total is 30  
Goodbye.
```

(10)

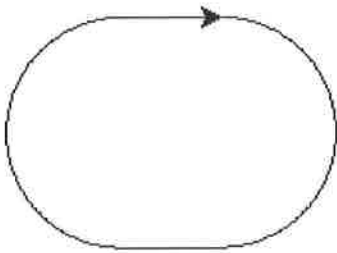
4) Using Python with turtle, draw the following shapes:

a) An oval. Remember that the code to draw a circle is:

```
import turtle

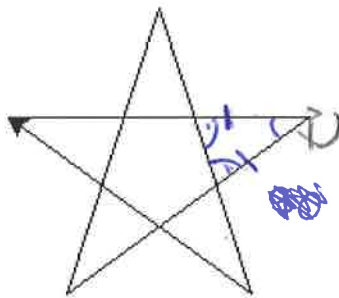
count = 0
while (count < 360):
    turtle.forward(1)
    turtle.right(1)
    count = count + 1
```

You should start with this as a base, and modify it to form an oval similar to:



(7)

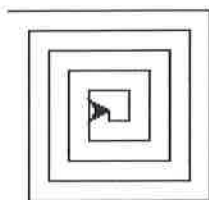
b) A 5 point star. This can be achieved without lifting the pen. It should look similar to:



Hint: the angle of rotation at the end of each point is 144 degrees.

(8)

c) A spiral. This may be a square spiral similar to:



You should not equate each edge to a line of code. Instead consider what you can use to reduce the amount of code you are required to produce.

(10)

test (3).py

#Sara Vonk

```
# .....
#          3) A
# .....
#print("hello")

#num1=0
#num2=0
#while num1!=-1:
#    num1=int(input("please enter a number(-1 to exit): "))
#    num2=num1+num2
```

```
#print("your total is {0}".format(num2))
#print("")
#print("Good Bye")
```

```
# .....
#          3) B
# .....

```

```
#print("hello")
```

```
#num1=0
#num2=0
#while num1!=-1:
#    num1=int(input("please enter a number(-1 to exit): "))
#    if (num1<-1):
#        print("no negative number")
#        num1=0
#    num2=num1+num2
```

```
#num2=num2+1
#print("your total is {0}".format(num2))
#print("")
#print("Good Bye")
```

✓ Good

```
# .....
#          4)
# .....

```

```
import turtle
```

```
#count=0
#while(count<180):
#    turtle.forward(1)
#    turtle.right(1)
#    count=count+1
```

```
#turtle.forward(100)
```

```
#count=0
#while(count<180):
#    turtle.forward(1)
#    turtle.right(1)
#    count=count+1
```

```
#turtle.forward(100)
```

```
## .....
#start
# .....

```

```
#turtle.forward(200)
#turtle.right(144)
```

test (3).py

```
#turtle.forward(200)
#turtle.right(144)
#turtle.forward(200)
#turtle.right(144)
#turtle.forward(200)
#turtle.right(144)
#turtle.forward(200)
```



```
#.....
#spiral fist one
#.....
```

```
#turtle.forward(100)
#turtle.right(90)
#turtle.forward(100)
#turtle.right(90)
#turtle.forward(100)
#turtle.right(90)
```

```
#turtle.forward(80)
#turtle.right(90)
#turtle.forward(80)
#turtle.right(90)
```

```
#turtle.forward(60)
#turtle.right(90)
#turtle.forward(60)
#turtle.right(90)
```



```
#turtle.forward(40)
#turtle.right(90)
#turtle.forward(40)
#turtle.right(90)
```

```
#turtle.forward(20)
#turtle.right(90)
#turtle.forward(20)
#turtle.right(90)
```

```
#.....
#spiral second
#.....
```



```
num1=200
while num1>0:
    num1=num1-20
    turtle.forward(num1)
    turtle.right(90)

    turtle.forward(num1)
    turtle.right(90)
```

