

Ceri Puddle - Marks

Q1 a) Algorithm for tea: 17 /17

b) Different drink selections - 1/5

c) Storing data on a computer - 3/3 **Total for Question 1 - 21 /25**

Q2 Flowchart /25 **Total for Question 2 - 20 /25**

Q3 a) Python code for 2) above 15/15

b) (Advanced) - 10/10 **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 - 91%



Mark agreed

91%

Computational Thinking Test Oct 2017

Name ...Ceri...puddle...

Student number ...150771...

Programme ...Bsc computer networks...

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)

making a cup of milky coffee with 1 teaspoon of sugar

- Take kettle to sink
- open lid and fill with cold water
- Close lid and plug kettle into socket
- Turn socket on and turn on the kettle switch.

Start

- 1 Take Kettle to Sink
- 2 open Lid
- 3 fill with cold water
- 4 close Lid
- 5 plug Kettle into Socket
- 6 Turn on socket
- 7 Turn Switch to on for the Kettle

8. pick up large mug
9. pick up 1L carton of milk
10. pour milk into cup
11. fill to 1/3 full
12. put down milk
13. put down mug
14. open up Jar of instant coffee
15. Take a tea spoon
16. Take 1 tea spoon of coffee from Jar
17. put into mug

(PTO for more space)

18. put lid back on jar of coffee
19. open bag of sugar ✓1
20. take tea spoon
21. Take 1 tea spoon of sugar
22. put in ~~tea~~ mug ✓3
23. put down tea spoon
24. pick up kettle when fully boiled
25. pour till full into mug ✓2
26. put kettle down +2
27. stir mug with tea spoon ✓1
28. enjoy

(17)

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.

When the program is started the user must select a drink before the robot is permitted to make anything. This can be done by introducing an input that would then trigger either a loop of a set of if statements. When the coffee has been completed the loop will go back to the input waiting for user interaction. (1)

(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? ✓2

The data would be stored in variables selected by an user input and would stay there until they are cleared or the program has been terminated. ✓1

(3)

(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.

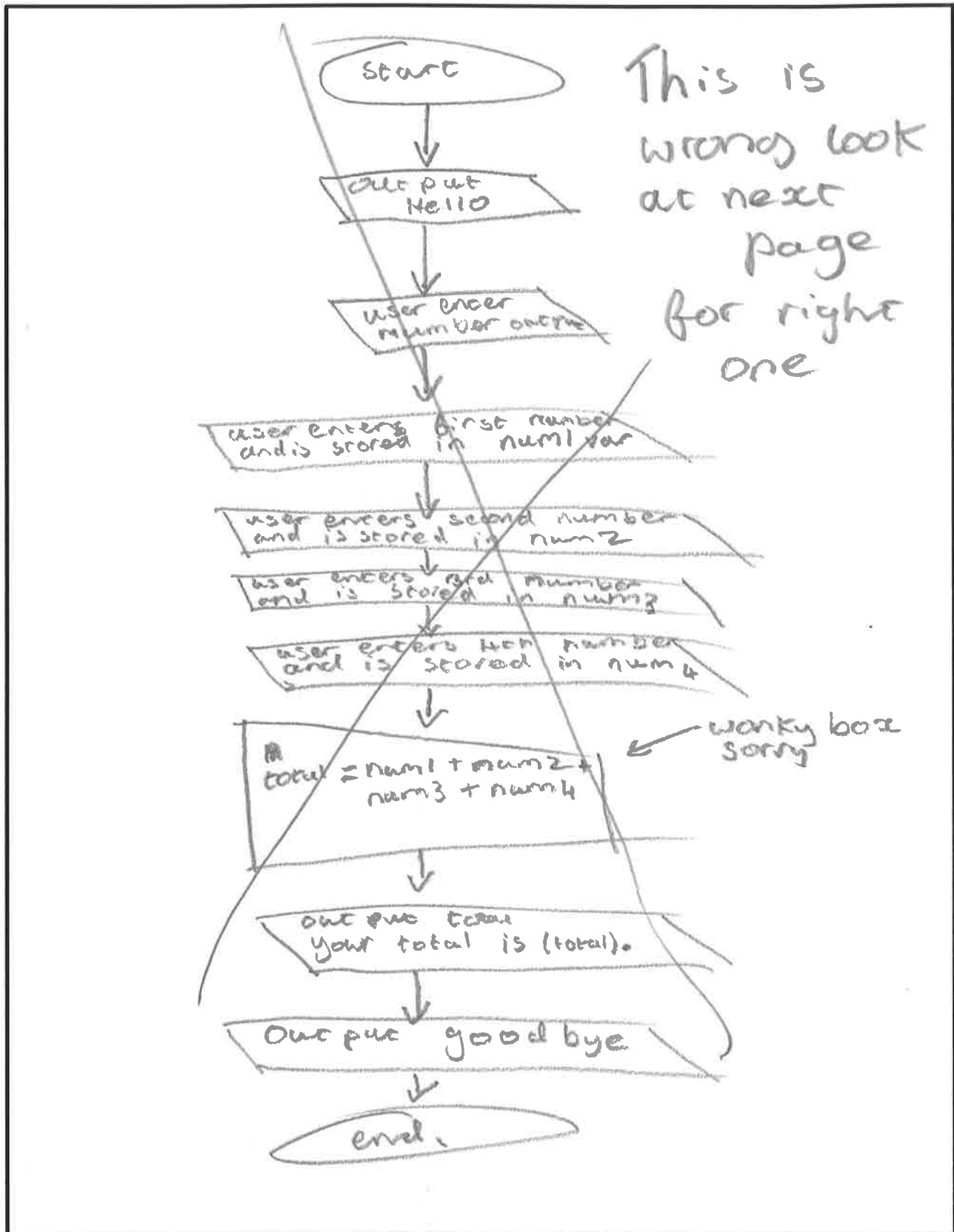
For example, the program may run something like:

```
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:



```
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.
```

```
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.
```

```
graph TD
    Start([start]) --> OutputHello[output Hello]
    OutputHello --> InputNumber{user input number}
    InputNumber --> AddTotal1[add to total]
    AddTotal1 --> IsLessOrEqual10{is less than or equal to 10}
    IsLessOrEqual10 --> AddTotal2[add to total]
    AddTotal2 --> IsLessOrEqual10
    IsLessOrEqual10 --> OutputTotal[output total is 10]
    OutputTotal --> GoodbyeOutput[good bye output]
    GoodbyeOutput --> End([end])
```

Handwritten flowchart for a program that calculates the sum of numbers from 1 to 10. The flowchart starts with an oval 'start', followed by a rectangle 'out put Hello', then a diamond 'user input number'. A loop structure follows: a rectangle 'add to total', a diamond 'is less than or equal to 10', and another rectangle 'add to total'. The loop continues until the condition is false, leading to a rectangle 'out put total is 10', then a rectangle 'good bye output', and finally an oval 'end'. Red checkmarks and numbers are placed next to various steps.

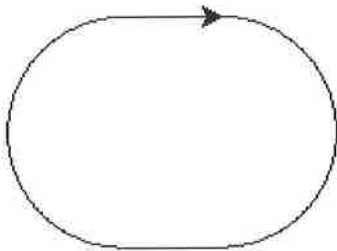
4) Using Python with turtle, draw the following shapes:

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

```
import turtle

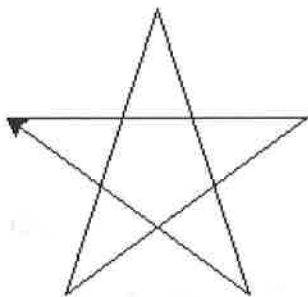
count = 0
while (count < 360):
    turtle.forward(1)
    turtle.right(1)
    count = count + 1
```

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



(7)

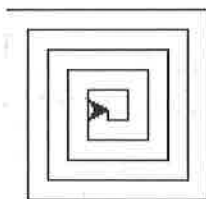
b) A 5 point star. This can be achieved without lifting the pen. It should look similar to:



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)

% pointed star.py

```
##Ceri Puddle p150771
import turtle
count = 0
while (count <= 4):
    turtle.forward(100)
    turtle.right(144)
    count = count +1
```

Enter a number.py

```
#Ceri Puddle p150771  
print ("Hello")
```

```
num = 0  
total = 0
```

```
while (num != -1):  
    num = int(input("Enter Number (-1 to exit) :"))  
    if (num <= -2):  
        print ("You can not enter negative number")  
    else:  
        total = num + total  
else:  
    print("Your total is {}".format(total))  
    print("Goodbye!")
```

✓ *Good*

oval (2).py

```
##Ceri puddle p150771  
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)
```

spiral (1).py

```
##Ceri Puddle p150771
import turtle
count = 0
forward = 200
while (count <= 19):
    turtle.forward(forward)
    turtle.right(90)
    count = count + 1
    forward = forward - 10
```

