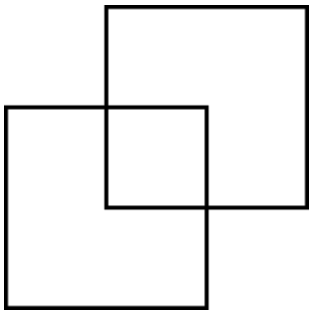
COMP0015: Introduction to Programming T1 (23/24)

Sample Test Nov/2023

Name and ID: _____

1. DURATION: 1 HOUR
2. Do not remove this question sheet from the room.
3. Calculators are not allowed. Textbooks and other materials are not allowed.
4. Students may leave the room by raising their hand. They will not be allowed back into the room.
5. All phones must be turned off and there must be silence during the test.
6. This exam is out of 40 marks in total. Total time is one hour. The number of marks roughly corresponds to the difficulty of the problem. Good luck!

1. [4 Marks] (Splinter)



Write turtle code to draw the above overlapping squares. The squares should have side length of 100 pixels; the corner of each is centered within the other.

(additional room for "Splinter," if needed)

Name:

---

2. 4 Marks (Digital Products)

Listing 1 shows the code that defines function `mul(s)` where `s` is a string.

```
1  def mul(s):
2      res = 1
3      for i in range(0, len(s)-1, 1):
4          res *= int(s[i])
5
6      return res
```

Listing 1: Function mul.

What is the output for the following function calls:

1. `mul("3421")` _____
2. `mul("2180")` _____
3. `mul("357 a")` _____

(additional room for "Digital Products," if needed)

3. 4 Marks (Accounting)

Write a function `count` that accepts a string and a character as parameters and returns the number of occurrences of the character in the string. Note that, the comparison between two characters is case insensitive; for example, 'a' and 'A' are considered the same.

Here are some example calls to the function and expected return results:

Function call	Value returned
`count("Good morning!", 'g')`	2
`count("Good morning!", 'h')`	0
`count("@Good morning!  * 2", '2')`	1

(additional room for "Accounting," if needed)

4. 6 Marks (Multiple sums)

   If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

   Write a program that reads a positive integer n and outputs the sum of all the multiples of 3 or 5 below n.

(additional room for "Multiple Sums," if needed)

5. 7 Marks (Cerberus)

Write a function called `three_heads` that repeatedly flips a coin until 3 heads are thrown. Each time a coin is flipped, the function will print H for heads or T for tails. You will use the `random.randint()` function to generate a value for either heads or tails. When 3 heads in a row are thrown, the function should print a message congratulating the user and should terminate. Here is one possible output when calling the function:

```
H H T H H T H T T H T T T H H H
Congratulations! 3 heads in a row.
```

(additional room for "Cerberus," if needed)

Name:

6. 7 Marks (Scrooge)

Write a function called `get_coins` that calculates the coins that represent a given amount that is always less than 100 pence. The function `get_coins` has one parameter, `amount`, which is the amount for which you will calculate the coins needed. The function will print the number of 50p, 20p and 10p coins required. There may be a remainder, your function should also print the remainder. Here is the expected output when calling the function with different values for `amount`:

```
Amount is: 0
remainder 0

Amount is: 99
50p - 1
20p - 2
remainder 9

Amount is: 9
remainder 9

Amount is: 11
10p - 1
remainder 1

Amount is: 54
50p - 1
remainder 4
```

(additional room for “Scrooge,” if needed)

7. [8 Marks] (Watch Time)

Write a function `enough_time` that takes four integers hour1, minute1, hour2, and minute2 as parameters. Each pair of parameters represents a time on the 24-hour clock (for example, 15:45 PM would be represented as 15 and 45). The function should return True if the gap between the two times is long enough to do some revision: that is, if the second time is at least 45 minutes after the first time. In all other cases, the function should return False. The function must also return the duration in minutes of the interval between the start and end times. You may assume that all parameter values are valid: the hours are both between 0 and 23, and the minute parameters are between 0 and 59. You may also assume that both times represent times in the same day. If the second time is earlier than the first time; your function should return (False, 0).

Here are some example calls to the function and expected return results:

Function call	Value returned
`enough_time`(11, 00, 11, 59)	(True, 59)
`enough_time`(12, 30, 13, 00)	(False, 30)
`enough_time`(12, 30, 13. 15	(True, 45)
`enough_time`(14, 20, 17, 02)	(True, 162)
`enough_time`(12, 30, 9, 30)	(False, 0)
`enough_time`(12, 00, 11, 55)	(False, 0)

(additional room for "Watch Time," if needed)