

THE AUSTRALIAN NATIONAL UNIVERSITY
Mid Semester Examination – September 2016

COMP1730 / COMP6730
Programming for Scientists

Study Period: 15 minutes

Time Allowed: 2 hours

Permitted Materials: One A4 page (1 sheet) with handwritten notes on both sides. NO calculator permitted. Use a blue or black pen.

Questions are NOT equally weighted. The questions and each of their parts are NOT in order from easy to hard.

Some questions or parts of questions are labelled “COMP6730 students only”. These should only be answered by students enrolled in COMP6730.

*For **COMP1730 students**, the exam will be marked out of 20, and worth 20% of the final course mark.*

*For **COMP6730 students**, the exam will be marked out of 24, and worth 20% of the final course mark.*

Where not otherwise indicated, questions are followed by framed blank panels into which your answers are to be written. If the space in the answer panel is not enough, use the additional panels at the end of the exam paper. If you use an additional panel, make sure you specify which question and part it is for.

Please write and express yourself clearly – if we cannot read what you have written or understand you have tried to say, your answer will not be considered correct.

Student Number (NOT your name): 	Course (write "1730" or "6730"):
---	--

The following are for use by the examiners.

Q1	Q2	Q3	Q4	Q5	Q6	Total
----	----	----	----	----	----	-------

Question 1 [7 marks]

- (a) Which of the following names are valid function names in python? (Write your answer, 'yes' or 'no', in the table.)

Name	Yes or No?	Name	Yes or No?
<code>__abc_def__</code>		<code>AFloat</code>	
<code>my-func</code>		<code>do2times</code>	
<code>a+_b</code>		<code>important</code>	
<code>"abc def"</code>		<code>42nd_char</code>	

[1 mark]

- (b) For each of the following expressions what is the type of the resulting value? (Write your answer in the table.)

Expression	type	Expression	type
<code>42/2</code>		<code>['1', '0', '0']</code>	
<code>11*"1"</code>		<code>21<42 + 42>21</code>	
<code>str(21 + 0.42)</code>		<code>"true", False</code>	
<code>[[]]</code>		<code>[1, 2] + [3 + 4]</code>	

[2 marks]

- (c) For each of the following expressions, write the value that it evaluates to into the table.

Expression	value	Expression	value
<code>-1 ** 2</code>		<code>chr(ord('A') + 3)</code>	
<code>4 % 15</code>		<code>len("namespace"[4:])</code>	
<code>5 / 5 * 100</code>		<code>[[1] + [2 + 3]]</code>	

[2 marks]

- (d) Suppose `s = 'Quam bene non Quantum'`. For each of the following expressions, write the value that it evaluates to into the table. (Don't forget that the letter case matters, and that string values must be enclosed in quotation marks.) If the expression results in an error, you only have to write "error".

Expression	value	Expression	value
<code>s[-7:-2]</code>		<code>s[5:7] + s[-3] + s[2]</code>	
<code>s[-len(s):1]</code>		<code>s[10:13].upper()</code>	
<code>s[len(s)]</code>		<code>s[s.find('o'):s.find('p')]</code>	

[2 marks]

Question 2 [3 marks]

Each of the following pieces of python code attempt to print the ratio $(a + b)/a$. For each one, indicate whether it is correct; if it is not, explain *precisely* what is wrong with it. (For example, if it has a syntax error, describe which line, or part of a line, is incorrect; if it runs but prints the wrong value, give a concrete example of inputs and output.) Note that any number (zero, one or two) of them may be correct. Assume `a` and `b` are defined in the global namespace, and have numeric values.

(a)

```
def sum(a, b):
    return a + b

def ratio(total = sum(a, b), div):
    return total / div

print("The ratio is", ratio(a, b))
```

(b)

```
def sum(a, b):
    return a + b

def ratio(a, b):
    return a / b

print("The ratio is", ratio(sum(a, b), a))
```

(c)

```
def sum(a, b):
    return a + b

def ratio(a, b):
    sum(a, b) = a + b
    return sum(a, b) / a

print("The ratio is", ratio(a, b))
```

Question 3 [3 marks]

(a) Two functions are defined as follows:

```
def funA(x):
    print("A: x = ", x)
    return 2*x

def funB(y):
    print("B: y = ", y)
    return funA(y) + 1
```

What is *printed* when the statement `print(funB(2 + funA(1)))` is executed? (You must write down all output, in the correct order. Make sure you clearly indicate where there are line breaks.)

[1 mark]

(b) A function `funC` is defined as follows:

```
def funC(a, b):
    k = a
    total = 0
    while k <= b:
        total = total + 1/k
        k = k + 1
    return 1/(b - a) * total
```

For each of the following calls to the function, indicate whether a runtime error will occur, the function will enter an infinite loop, or if it will return a value. (You don't have to calculate the exact value it returns.)

(i) `funC(-2, 2)`
(ii) `funC(2, 0)`

[2 marks]

Question 4 [5 marks]

(a) What are the values of variables **a**, **b**, **c** and **d** after the following code is executed:

```
a = [6,5,4]
b = [3,2,1]
c = b[1:]
d = b
b.extend(c)
d.append(a)
a.sort()
```

a =	
b =	
c =	
d =	

[2 marks]

(b) What is printed by lines #1, #2 and #3 when the following code is executed? (If an error occurs before all three lines have executed, write the output of the lines before and then "error".)

```
def f(a, b):
    if len(b) > 0:
        x = a + b
        b.pop(-1)
    return x.pop(0)
```

```
x = [3,2,1]
y = [4,5]
print(f(x, y)) # line 1
print(f(x, y)) # line 2
print(f(x, y)) # line 3
```

Line 1:	
Line 2:	
Line 3:	

[2 marks]

(c) Strings (**str**), lists (**list**) and tuples (**tuple**) are python's built-in sequence types. Describe one property that strings and tuples have in common, but lists do not have; and one property that lists and tuples have in common, but strings do not have.

[1 mark]

Question 5 [COMP1730: 2 marks; COMP6730: 4 marks]

The following function takes as argument a sequence:

```
def funX(seq):
    if len(seq) == 1:
        return seq[0]
    elif seq[0] > funX(seq[1:]):
        return seq[0]
    else:
        return funX(seq[1:])
```

- (a) What does the function return when called with the argument [2,3,1]?

[1 mark]

- (b) Explain in plain English what this function does *in general*. Make your explanation as general and informative as you can. A good answer is one that describes the purpose of the function – something you might put into a comment or docstring – *not* a line by line description of how it works.

[1 mark]

- (c) [COMP6730 students only] Write a *non-recursive* function that for any argument of a sequence type returns the same value as `funX`.

[2 marks]

Student Number:

Question 6 [COMP6730 students only: 2 marks]

The complexity of deciding if a given value, x , occurs in a list of n elements can be done by iterating through the list and comparing each element in it with x . In the worst case, this requires examining every element in the list once. The problem cannot be solved with, in the worst case, fewer comparisons than this. Thus, the time complexity of this problem is $O(n)$.

What is the complexity of deciding if a given value occurs at least k times in a list of n elements, where k is any non-negative integer?

Give an informal proof both that there is an algorithm with the time complexity of your answer, and that it cannot be done in less time.

Student Number:

Empty rectangular box for student response.

Empty rectangular box for student response.

Empty rectangular box for student response.

Empty rectangular box for student response.