

THE AUSTRALIAN NATIONAL UNIVERSITY  
Mid Examination – September 2017

**COMP1730 / COMP6730**  
**Programming for Scientists**

*Study Period: 15 minutes*

*Time Allowed: 2 hours*

*Permitted Materials: One A4 page (1 sheet) with 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. Students in COMP1730 will not receive any marks for answering these.*

*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 (6)	Q2 (4)	Q3 (3)	Q4 (2/3)	Q5 (3/4)	Q6 (2/4)	Total (20/24)

**Question 1 [6 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>**result**</code>		<code>function</code>	
<code>_type</code>		<code>IMPORT</code>	
<code>0th_int</code>		<code>'delta'</code>	
<code>zero:th_int</code>		<code>int</code>	

[1 mark]

- (b) For each of the following expressions, write the value that it evaluates to, and the type of that value, into the table. If evaluating the expression results in a runtime error, you only have to write "error".

Expression	value	type
<code>2 - 2 / 4</code>		
<code>25 // 5 * 5</code>		
<code>int(4) / 2</code>		
<code>int('2' + '0')</code>		
<code>-1 ** -2 == 1</code>		
<code>'1' * 0 + '1'</code>		
<code>"1 // 3"</code>		
<code>(1 + 2) * [3 + 4]</code>		
<code>["AB"]</code>		

[3 marks]

- (c) Suppose `a = [[3,6,9], [2,5,8], [1,4,7]]`. For each of the following expressions, write the value that it evaluates to into the table. (If evaluating the expression results in an error, you only have to write "error".)

Expression	value	Expression	value
<code>a[1:2][1]</code>		<code>a[1][1:2]</code>	
<code>a[2][2:] + a[-2][:-2]</code>		<code>a[1:][0]</code>	
<code>a[-1][:1] * a[1][0]</code>		<code>a[-3][1] + a[1][-1]</code>	

[2 marks]

**Question 2 [4 marks]**

Each of the following pieces of python code attempt to print the value of the expression  $(a + b)/b$ . For each one, answer whether it is correct, meaning that it *runs without error and prints the correct value*. 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, two, three or four) of them may be correct. Assume that variables `a` and `b` have been set and have numeric values.

(a) 

```
def ratio(a, b):  
    a = a + b  
    print("The ratio is", a / b)
```

(b) 

```
def ratio(a, b):  
    return a + b / b  
  
print("The ratio is", ratio(a, b))
```

(c) 

```
def sum(a, b):  
    print(a + b)  
  
print("The ratio is", sum(a, b) / b)
```

(d) 

```
def ratio():  
    return ans = (a + b) / b  
  
print("The ratio is", ratio(a, b))
```

**Question 3 [3 marks]**

(a) A function `funA` is defined as follows:

```
def funA(a, b):  
    c = 0  
    while a != b:  
        if a % 5 == 0:  
            c = c + 1  
        elif a > b:  
            b = b + 1  
        else:  
            a = a + 1  
    return c
```

For each of the following calls to the function, answer whether the function will return a value, enter an infinite loop, or if or if a runtime error will occur. You do not have to calculate the exact value that the returns or name the specific error.

(ii) `funA(10, 0)`

(i) `funA(3, 7)`

[2 marks]

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

```
def funB(x):  
    return 2*x  
    return x + 1
```

What is returned by the call `funB(2)`?

[1 mark]

**Question 4 [COMP1730: 2 marks; COMP6730: 3 marks]**

(a) What is printed when the following python is run?

```
def funC(x):
    print("funC: x =", x)
    if x < 0:
        return funC(-2 * x)
    else:
        return x - 2

def funD(x, y):
    print("funD: x =", x, "y =", y)
    return funC(y) - funC(x)

print(funD(-2, funC(2)))
```

Write down each line that is printed, in the correct order.

[2 marks]

(b) [COMP6730 students only] Does the loop in the following python code terminate? Explain why or why not.

```
x = 1
y = 0
z = x / 10
while x != z:
    y = y + x
    x = z
    z = x / 10
print(y)
```

[1 mark]

**Question 5 [COMP1730: 3 marks; COMP6730: 4 marks]**

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

```
a = [2,1]
b = [a, a]
c = a[1:]
b[0] = c
d = b
d.append(c)
d[2].append(1)
b[1].sort()
```

a =	
b =	
c =	
d =	

[2 marks]

(b) Give one example of an operation that can be done on lists (values of type `list`) but not on tuples (values of type `tuple`), and one example of an operation that works on both tuples and lists. Your examples should include python code.

[1 mark]

(c) [COMP6730 students only] It has been mentioned in the lectures that there are data types in python that support iteration using a `for` loop, but which are not sequences, and can not be indexed. Suppose `x` is a value of such a data type; explain how you could use a `for` loop to find the `i`:th element in `x`, without indexing `x`. (You do not have to write python code.)

[1 mark]

**Question 6 [COMP1730: 2 marks; COMP6730: 4 marks]**

The following function takes as argument a sequence:

```
def funX(seq):
    count = 0
    index = 0
    while index < len(seq):
        if seq[-1 - index] == seq[index]:
            count = count + 1
            index = index + 1
    return count == len(seq)
```

- (a) What does the function return when called with the argument `[2,1,2,1]`? Specify both the return value and its type.

[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 docstring – *not* a line by line description of how it works.

[1 mark]

- (c) [COMP6730 students only] Write a function that for any argument of a sequence type returns the same value as `funX`, *without using a loop* (either `while` or `for`).

[2 marks]

Student Number: .....

Empty rectangular box for student information.

Empty rectangular box for student information.

Empty rectangular box for student information.

Empty rectangular box for student information.