

THE AUSTRALIAN NATIONAL UNIVERSITY  
Final (Part A) Examination – November 2017

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

*Study Period: 15 minutes*

*Time Allowed: 3 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 25, and worth 25% of the final course mark.*

*For **COMP6730 students**, the exam will be marked out of 30, and worth 25% 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 (5)	Q2 (4)	Q3 (4)	Q4 (4)	Q5 (5/6)	Q6 (1/3)	Q7 (2/4)	Total (25/30)

**Question 1 [5 marks]**

- (a) 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>0 % 2</code>		
<code>{0, 2} - {1, 2}</code>		
<code>2 in { 1 : 2 }</code>		
<code>set("Aachoo")</code>		
<code>3 &lt; 4 - 1 &lt; 4</code>		
<code>1, 2 &lt; 3</code>		
<code>list('1') + list('12')</code>		
<code>[[1], [2]][1][0]</code>		
<code>{}</code>		

[3 marks]

- (b) Suppose `a = ['ab', 'bcd', 'cde']`. 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[1:2][1]</code>	
<code>a[1][:-1]</code>		<code>a[-2] + a[2]</code>	
<code>a[:-2] + a[2:]</code>		<code>len(a[1:len(a[1]):])</code>	

[2 marks]

**Question 2 [4 marks]**

Each of the following pieces of python code attempt to compute and print the maximum difference between any pair of numbers in a list `x`. (Note that the maximum difference is always greater than or equal to zero, which is the difference between any number and itself.) 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 causes a runtime error or prints the wrong value, give a concrete example of inputs and output. Do *not* answer the question with how you think the code should have been written.) Note that any number (zero, one, two, three or four) of them may be correct. Assume that variable `x` is defined in the global namespace and that its value is a non-empty list of numbers.

```
(a) def find_max(x, y):
    c = []
    for i in x:
        for j in y:
            c.append(x[i] - y[j])
    return max(c)
print("The max difference is ", find_max(x, x))
```

```
(b) def find_max(x, y):
    d = 0
    for i in x and for j in y:
        d = max(d, i - j)
    return d
print("The max difference is ", find_max(x, x))
```

```
(c) def find_max(y):
    z = x[0] * y
    for i in range(len(x)):
        x[i] = y * x[i]
        if x[i] > z:
            z = x[i]
    return z
print("The max difference is ", find_max(1) + find_max(-1))
```

```
(d) def find_max(x):
    x = x.sort()
    return x[-1] - x[1]
print("The max difference is ", find_max(x))
```

**Question 3 [4 marks]**(a) A function `funA` is defined as follows:

```
def funA(x):
    i = 0
    while i < len(x):
        i = i + x[i] % len(x)
    return i
```

What is returned by each of the following calls to the function? If the call results in a runtime error, just write “error”. If it results in an infinite loop, just write “infinite loop”. If the function returns a value, write the return value exactly.

(i) `funA([1,3,2])`

(ii) `funA([2,3,1])`

**[2 marks]**(b) A function `funB` is defined as follows:

```
def funB(x):
    for y in x:
        if y % 2 == 0:
            return y - 1
        if y < 0:
            return 2 * y
```

What is returned by each of the following calls to the function? If the call results in a runtime error, just write “error”. If it results in an infinite loop, just write “infinite loop”. If the function returns a value, write the return value exactly.

(i) `funB([5,4,-3,-2,1])`

(ii) `funB([5,3,1])`

**[2 marks]**

**Question 4 [4 marks]**

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

```
def funF(y, a):
    z = y[a - 1]
    y.pop(a - 1)
    x = y + [ z ]
    print("F:", x)
    return x[:-1]

x = [2,3,1]
funF(funF(funF(x, 2), 1), 1)
print("x:", x)
```

Write down each line that is printed, in the correct order. If a runtime error occurs, write down all that is printed before the error, and then "error".

**[2 marks]**

(b) Consider the following python code:

```
def funE(x):
    if x < 0:
        y = 2
        return -x * y
    else:
        return x * y

x = -1
y = 1
```

After the code above has been executed, what is the value returned by each of the following function calls? If the function call causes a runtime error, just answer "error".

(i) `funE(2)`
(ii) `funE(-2)`**[2 marks]**

**Question 5 [COMP1730: 5 marks; COMP6730: 6 marks]**

- (a) Files, sets and dictionaries are all examples of what are called *iterable* types in python. Briefly describe what is the common characteristic of iterable types.

[1 mark]

- (b) The following function inverts a dictionary:

```
def invert_dict(A):  
    invA = {}  
    for key in A.keys():  
        invA[A[key]] = key  
    return invA
```

- (i) What runtime errors can occur in `invert_dict`? You don't have to write the exact name of each error type, but you must explain where and how it may occur. *Assume that the argument is a dictionary* (that is, "TypeError because the argument is not a dictionary" is not an acceptable answer).

- (ii) If `A` is a dictionary, is it always the case that `invert_dict(invert_dict(A)) == A`? (assuming no runtime error occurs). If yes, explain why. If no, give an example of an input dictionary that proves it.

[2 marks]

Student Number: .....

- (c) For each of the calls to `print` in the following code, write what will be printed into the box after the statement executes. (In some cases there are several possible outputs, because elements in sets and dictionaries have no order. In those cases, any output that could be printed is acceptable.) If any statement causes an error, assume that it is skipped and execution continues with the following statements (as if the statements were entered interactively).

```
a = { 'a' : 'b', 'b' : 'c', 'c' : 'b' }  
c = list(a.values())  
c.sort()  
print(c)
```

```
b = a.copy()  
for x in c:  
    a[x] = x + a[x]  
print(a)
```

```
d = set(b.values())  
print(d)
```

```
for x in d:  
    a[x] = 0  
print(a)
```

[2 marks]

- (d) [COMP6730 students only] The floating point number representation has limited precision. Give an example of a problem that can arise as a *consequences* of this limitation – that is, something that happens because of the limited precision – and explain what, if anything, you can do when writing code to avoid this problem.

[1 mark]

**Question 6 [COMP1730: 1 mark; COMP6730: 3 marks]**

We define an abstract data type called *trajectory* as a sequence of *points*. The key operation on values of this abstract type is to get the next point, from a given point, along the trajectory. Thus, its interface includes the function `next_point(point, traj)`, which returns the next point after `point` in `traj`.

You can assume that every point in the trajectory is unique (not equal to any other point), as otherwise it can be ambiguous which value should be returned by `next_point`. However, you should make no assumptions about how points are represented. A point could be a coordinate (e.g., [-35.3, 149.1, 1893]), a description (e.g. "corner of North rd and Daley rd") or anything else.

- (a) Describe one possible data structure that can be used to implement the abstract data type. You do not have to write code for an actual implementation, but you should describe how a trajectory is stored using this data structure.

[1 mark]

- (b) [COMP6730 students only] Write an implementation of the `next_point(point, traj)` function, assuming the data structure you proposed for the previous question. The function should return `None` if there is no next point after `point` in `traj`.

[1 mark]

- (c) [COMP6730 students only] What is the complexity of your implementation of the `next_point` function? Give the answer in big-O notation, along with a justification.

[1 mark]



**Question 7 [COMP1730: 2 marks; COMP6730: 4 marks]**

The following function takes two arguments:

```
def funX(a_list, x):
    a_list.sort()
    index = 0
    while a_list[index] < x:
        index = index + 1
    return a_list[index]
```

- (a) What does the function call `funX([1,5,3], 1.5)` return? Specify both the value and its type.

[1 mark]

- (b) Explain in plain English what the function `funX` does. 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] Give two examples of different runtime errors that can occur in `funX`, assuming the first argument is a list (that is, “`TypeError` if `a_list` is not a list” is not an acceptable answer). For each, describe how it can occur (for example, give an example of inputs that will cause the error).

[1 mark]

- (d) [COMP6730 students only] For each of the two runtime errors in your answer to question (c), explain whether this error should be caught (using a `try-except` statement), and if so, how it should be handled, or whether an `assert` statement should be added to prevent that error.

[1 mark]

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.