

Student Number

The University of Melbourne  
Department of Computing and Information Systems

**Mid-semester Test, Semester 1, 2014**  
**COMP10001 Foundations of Computing**

**Writing Time:** 45 minutes

This paper has 8 pages including this cover page.

There are 5 questions in the paper, for a total of 45 marks (one mark for each minute of test time).

- All questions should be answered by writing a brief response or explanation in the lined spaces provided on the examination paper.
- It is not a requirement that all the lined spaces be completely filled; answers should be kept concise.
- Only material written in the lined spaces provided will be marked.
- Your writing should be clear; illegible answers will not be marked.
- Extra space is provided at the end of the paper for overflow answers. Please indicate in the question you are answering if you use the extra space.
- Your answers can use any of the standard Python libraries, but be sure to call/import them correctly.

**Authorised Materials:** No materials are authorised.

**Calculators:** Calculators are not permitted.

<i>Examiners' use only</i>						
1	2	3	4	5	Total	

**Question 1****[4 marks]**

Evaluate the following expressions; what is the “return value” in each case?

$1/2 + 1$

---

`sorted({'b':2, 'a':1}.items())`

---

`bool("winnie" and "balloon" or False)`

---

`['a', 'man', 'a', 'pan'][1][-1]`

---

**Question 2****[6 marks]**

Answer the following questions based on the following function, and the indicated line numbers:

```
1 def fun(a):  
2     for i in range(len(a)):  
3         if a[i] < 'a' or a[i] > 'z':  
4             return False  
5         if a[i] in a[:i]:  
6             return False  
7     return True
```

(a) What is line 3 testing for?

---

---

---

(b) What is line 5 testing for?

---

---

---

(c) What is the overall purpose of the function?

---

---

---

---

---

**Question 3****[5 marks]**

What is the output of the following code:

```
battleships = [['0','p','0','s'],  
               ['0','p','0','s'],  
               ['p','p','0','s'],  
               ['0','0','0','0']]
```

```
def fun(a,b,bships):  
    c = len(bships)  
    return bships[c-b][a-1]
```

```
print(fun(1,1,battleships))  
print(fun(1,2,battleships))
```

---

---

---

---

**Question 4****[8 marks]**

Given the following:

```
def coverage(    max = intlist[0]  
    min = intlist[0]  
    sum = max  
    for i in           
        if i < min:  
              
        elif i > max:  
            max = i  
    
```

supply code fragments to insert in each of the numbered boxes to complete the function `coverage`. `coverage` takes a single argument `intlist` in the form of a list of integers, and returns the average of the values, excluding the largest and smallest values in the list. You may assume that the list contains at least three integers. Example outputs of the function are:

```
>>> coverage([1, 3, 2, 100])  
2.5  
>>> coverage([1, 1, 3, 100])  
2.0  
>>> coverage([1, 1, 100])  
1.0
```

- (1) \_\_\_\_\_
- (2) \_\_\_\_\_
- (3) \_\_\_\_\_
- (4) \_\_\_\_\_
- (5) \_\_\_\_\_

**Question 5****[12 marks]**

Write code to implement each of the following functions:

- (a) `substr`, which takes string arguments `sup` and `sub`, and returns `True` if `sub` occurs in `sup`, and `False` otherwise. For example:

```
>>> substr("woolloomooloo", "oo")
True
>>> substr("woolloomooloo", "look")
False
>>> substr("cool", "oo")
True
```

---

---

---

---

---

---

- (b) `substrn`, which takes string arguments `sup` and `sub`, and returns the integer count of the (possibly overlapping) number of times `sub` can be found in `sup`. For example:

```
>>> substrn("woolloomooloo", "oo")
4
>>> substrn("woolloomooloo", "look")
0
>>> substrn("cool", "oo")
2
```

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**Question 6****[10 marks]**

Write a function `minfreq` that takes two arguments — `strlist`, a list of strings, and `minfreq`, an integer — and returns the number of distinct words in `strlist` which occur at least `minfreq` times. Note that your function shouldn't change the casing or remove punctuation from any of the strings in `strlist`. Example outputs of the function are:

```
>>> minfreq(["ace", "ball", "ace", "cot"], 4)
0
>>> minfreq(["ace", "ball", "ace", "cot"], 1)
3
>>> minfreq(["ace", "ball", "ace", "cot"], 2)
1
```

DRAFT

8 of 8