THE UNIVERSITY OF WESTERN AUSTRALIA

Computer Science and Software Engineering

SEMESTER 1, 2017 EXAMINATIONS

CITS1401
Problem Solving and Programming

FAMILY NAME: _____ GIVEN NAMES: _____

STUDENT ID: | | | | | | | | | SIGNATURE: _____

This Paper Contains: 9 pages **(including the title page)**
Time allowed: **2:00 hours**

INSTRUCTIONS:

- This examination paper comprises 1 section containing 8 questions. Answer all questions.
- Answers to all the questions are to be written in the spaces provided in this exam booklet.
- The exam is scored out of 120 marks. It may help you to think of it as (very roughly) a mark a minute.
- Use the blank pages at the beginning and end for rough work.
- No calculators or other aids can be brought into the exam

Office Use Only

Q1	Q2	Q3	Q4	Q5	Q6	Q6	Q8

PLEASE NOTE

Examination candidates may only bring authorised materials into the examination room. If a supervisor finds, during the examination, that you have unauthorised material, in whatever form, in the vicinity of your desk or on your person, whether in the examination room or the toilets or en route to/from the toilets, the matter will be reported to the head of school and disciplinary action will normally be taken against you. This action may result in your being deprived of any credit for this examination or even, in some cases, for the whole unit. This will apply regardless of whether the material has been used at the time it is found.

Therefore, any candidate who has brought any unauthorised material whatsoever into the examination room should declare it to the supervisor immediately. Candidates who are uncertain whether any material is authorised should ask the supervisor for clarification.

Supervisors Only – Student left at:

This page has been left intentionally blank

1. Write a Python function `splitEmailAddress(Address)`, which, given an email address, e.g. `"Michael.Wise@uwa.edu.au"`, returns a list of the component strings, e.g. `['Michael', 'Wise', 'uwa', 'edu', 'au']`. **[5 Marks]**

2. Write a Python definition for the function `basename(P)` which, given a pathname to a file (a string), returns a string with the file name without the preceding directories and without any suffix, if one exists. For example `basename('/Users/michaelw/CITS1401/exam.doc')` will return the string `'exam'`. (There is a function called `basename` in the `os.path` library, but please ignore it and instead use string processing functions.) **[5 Marks]**

3. The Manhattan distance between two points *(x1, x2)* and *(y1, y2)* – the distance a car needs to travel between two points in a city on a grid – is (in 2 dimensions):

$$manhat = |(y_1 - x_1)| + |(y_2 - x_2)|$$

Write a definition for `manhat(x, y)`, where `x` and `y` are points in N-dimensional space, each represented as a list of floating point values, e.g. `manhat([1, 3, 5, 7], [1, 9, 25, 42])` returns `61`. You can assume that the lists are the same length, though not necessarily length 4. (|..| stands for the absolute value function.) **[10 Marks]**

4. Consider the following rather impenetrable (but correct) Python code, taken from a function:

```
if os.path.exists(f):
   return [c for c in open(f,'r').read().split('\n') if c != ‘’]
```

   a. What does the code do? **[3 marks]**
   b. Rewrite the code so that it is easier to understand **[7 Marks]**.

a.

NOT COVERED

b.

5. What is seen as the result of executing the following Python code **[10 Marks]**:

```
def ft6b():
    numberGames = {}
    numberGames[(1,2,4)] = 5
    numberGames[(4,2,1)] = 10
    numberGames[(1,2)] = 12

    try:
        sum = 1
        for k in numberGames:
            sum *= numberGames[k]
        numberGames.append(sum)
        print('Try block executed.')
    except:
        print('Exception occured')
    print(numberGames, 'Sum=',sum)
```

NOT COVERED

6. Write a definition for the function `merge(list1, list2)`, that, given two lists: `list1` and `list2`, which are sorted in ascending order, returns a list that combines the two lists in ascending order, e.g. `merge([1,3,5,11,12], [2,4,6,8])` returns `[1,2,3,4,5,6,8,11,12]`. (Hint: For starters, you will need a while loop that compares the smallest item in each list. Please do not use any of Python's sorting function; apart from anything else, that is very inefficient in this context) **[30 Marks]**

7. Write a definition for the function, `marksdistribution(D)`. The input to `marksdistribution` is a dictionary mapping student names to marks in the range 0..100. The output from `marksdistribution` should be a dictionary mapping marks ranges seen at UWA, to counts of students from `D` with marks in the respective ranges. The definitions of the marks ranges are:
    - N < 50
    - P ≥ 50 and < 60
    - Cr ≥ 60 and < 70
    - D ≥ 70 and < 80
    - HD ≥ 80

    For example, if `D = {"Fred":55, "James":67, "Jemima":71}`, `marksdistribution(D)` will return a dictionary resembling `{"P":1, "Cr":1, "D":1}` **[30 Marks]**

8. Define a definition for the function, `pow(x, N)`, to compute x^N for integer x and integer N, e.g. 3^{1001}. (Large numbers will require *long* integers.) For large N, a recursive function can be more efficient than multiplying x N times, so a recursive function will be awarded more marks. Specifically, if you create a function that uses repeated multiplication it will be awarded a maximum of **[10 marks]**. However, if you write a recursive function, your solution will be marked out of **[20 marks]**. Hint (for recursive solution): What happens if you divide N by 2, i.e. first solve `pow(x, N//2)`.

---END OF EXAMINATION PAPER--