THE UNIVERSITY OF WESTERN AUSTRALIA
Achieve International Excellence

Computer Science and Software Engineering

SEMESTER 1, 2012 EXAMINATIONS

CITS1401
Problem Solving and Programming

FAMILY NAME: ______________________________ GIVEN NAMES: ________________________

STUDENT ID: | | | | | | | | | SIGNATURE: __________________________

This Paper Contains: **12** pages **(including title page)**
Time allowed: **2 hours 10 minutes**

INSTRUCTIONS:
Answer all questions. The marks for the paper total 100.
Write your answers in the spaces provided on this question paper.
No other paper will be accepted for the submission of answers.
Do not write in this space.

PLEASE NOTE

Supervisors Only – Student left at:

This page has been left intentionally blank

Q1. Arithmetic

A point on a plane can be represented as a pair of numbers *px, py* denoting its Cartesian coordinates; a circle can be represented as a triple of numbers *cx, cy, r*, denoting the coordinates of its centre and its radius. You may assume that the module *math* has been imported.

(a) Define the following function. **3 marks**

def area(cx, cy, r):
#area returns the area of the circle cx, cy, r

(b) Define the following function. **3 marks**

def inside(px, py, cx, cy, r):
#inside returns True if the point px, py is inside the circle cx, cy, r, and False otherwise

(c) Define the following function. **4 marks**

def half(cx, cy, r):
#half returns a tuple of three numbers representing a circle that lies inside
#the circle cx, cy, r, that has half its radius, and that touches it at its lowest point

Q2. Booleans and testing

(a) Define the following function. **3 marks**

def mid(x, y, z):
#mid returns the median value of x, y, z

e.g. *mid(1,3,2) = 2*, and *mid(2,1,2) = 2*.

(b) Define the following function. **7 marks**

def test_mid():
#test_mid returns True if mid is correct, and False otherwise

test_mid should perform all tests required to establish the correctness of *mid*.

Q3. List iteration

Pascal's triangle is an infinite structure as follows.

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
  …
```

Each non-edge number is the sum of the two numbers immediately above it.

(a) Use iteration over lists to define the following function. **5 marks**

def nextrow(xs):
#nextrow takes a list of numbers xs holding a row of Pascal's triangle,
#and it returns a list holding the row that follows xs

e.g. *nextrow([1,2,1]) = [1,3,3,1]*.

(b) Use iteration over lists to define the following function. **5 marks**

def pascal(n): # assume n > 0
#pascal returns a list of lists holding the first n rows of Pascal's triangle

e.g. *pascal(4) = [[1], [1,1], [1,2,1], [1,3,3,1]]*.

Q4. List comprehensions

(a) Use a list comprehension to define the following function. **3 marks**

def nomods(n, k): # assume n >= 0, k > 0
#nomods returns a list containing the numbers in 1.. n inclusive that are divisible by k

e.g. *nomods(10, 3) = [3, 6, 9].*

(b) Use a list comprehension to define the following function. **3 marks**

def everyk(xs, k): # assume k > 0
#everyk returns a list containing the first and then every k^th^ element from the list xs

e.g. *everyk([3,1,4,1,5,9], 2) = [3,4,5]*.

(c) Use a list comprehension to define the following function. **4 marks**

def square(n): # assume n >= 0
#square returns a square list of length n where the k^th^ row contains
#the first n multiples of k

e.g. *square(3) = [[1,2,3], [2,4,6], [3,6,9]]*.

Q5. Dictionaries

An index into a string is a data structure that tells you where each character in the alphabet occurs in the string. So given the string “Lyndon While is Welsh”, the data structure would store the fact that “W” occurs at indices 7 and 16, “h” occurs at indices 8 and 20, etc. Any character which doesn’t occur in the string doesn’t get an entry in the data structure.

Use the dictionary method *get* to define the following function. **10 marks**

def makeindex(s):
#makeindex returns a dictionary that holds an index for s

e.g. makeindex(“abba a-gogo”) =
{“a”: [0, 3, 5], “b”: [1,2], “g”: [7, 9], “o”: [8, 10], “-“: [6], “ “: [4]}.

Q6. String processing

(a) Use slicing and the string method *find* to define the following function. **3 marks**

def before1st(x, s): # assume x occurs in s
#before1st returns a string containing the part of s before the first occurrence of x

e.g. *before1st("l", "Wales") = "Wa"*.

(b) Use slicing and the string method *find* to define the following function. **7 marks**

def splitat2nd(x, s): # assume x occurs at least twice in s
#splitat2nd returns two strings, containing the parts of s before and after
#the second occurrence of x

e.g. *splitat2nd("n", "LyndonW") = ("Lyndo", "W")*.

Q7. Recursion over integers

When climbing a set of stairs, you can ascend either one or two steps with each stride. So for example, if you have to climb three stairs there are three different ways you could do it: touching every step; missing the first step; or missing the second step.

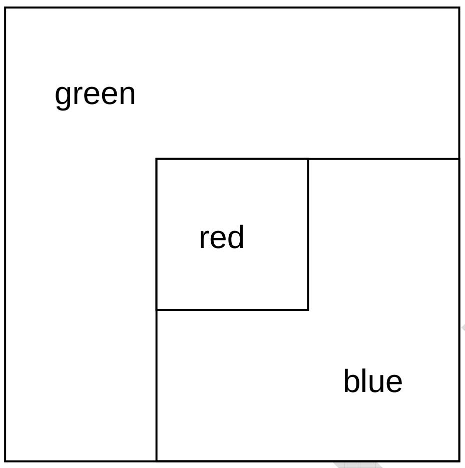
Use recursion to define the following function. **10 marks**

def stairs(n): # assume n > 0
#stairs returns the number of different ways that you can climb a set of n stairs

e.g. *stairs(4) = 5*, and *stairs(6) = 13*.

Q8. Turtle Graphics

The Noland national flag looks like this:



The flag is square, and the internal squares are 1/3 and 2/3 of the width of the flag.

Use Python's turtle graphics to define the following function. You may assume that the module *turtle* has been imported. **10 marks**

def noland(n): # assume n > 10
#noland draws the Noland national flag with width n using turtle graphics

Q9. Enumeration and search

(a) Describe the basic principles and efficient operation of the problem-solving technique “enumeration and search”. **5 marks**

(b) Illustrate your answer to (a) with a problem that is amenable to this technique, and sketch a solution to this problem that uses the technique. **5 marks**

Q10. Divide-and-conquer

(a) Describe the basic principles and efficient operation of the problem-solving technique "divide-and-conquer". **5 marks**

(b) Illustrate your answer to (a) with a problem that is amenable to this technique, and sketch a solution to this problem that uses the technique. **5 marks**

END OF PAPER