Durham University

EXAMINATION PAPER

Examination Session: May/June	**Year**: 2017	**Exam Code**: COMP1051-WE01

Title: Computational Thinking

Time Allowed:	2 Hours	
Additional Material provided:		
Materials Permitted:		
Calculators Permitted:	Yes	Models Permitted: Casio fx-83 GTPLUS or Casio fx-85 GTPLUS
Visiting Students may use dictionaries:	Yes	

Instructions to Candidates:	Answer THREE questions. (TWO from Section A and ONE from Section B.)

Revision:	

Computational Thinking (COMP1051-WE01)

Section A Computational Thinking (Prof. Iain A. Stewart)

Question 1

(a) Give 4 different functions of the operating system. **[4 Marks]**

(b) Explain what an interrupt is and how an operating system uses interrupts to deal with peripheral devices that operate much more slowly than the CPU. **[6 Marks]**

(c) What is a process, within an operating system, and what are the two essential elements of a process? Is it the case that an operating system for a CPU with one processor can only have one process? (Explain your answers and define any concepts used.) **[6 Marks]**

(d) Explain the principle of mutual exclusion, when we have two processes sharing the same memory location. Why do we need mutual exclusion? Give an illustration to show that ensuring mutual exclusion is important. **[6 Marks]**

(e) Explain how we describe the life-cycle of a process. (You need not give the full description; just explain **how** we describe the life-cycle.) **[3 Marks]**

continued

Question 2

(a) What is the Graph Colouring decision problem and what is the Graph Colouring optimisation problem? **[5 Marks]**

(b) Outline a greedy algorithm to colour a graph where your algorithm should try and use as few colours as possible. Show how your algorithm proceeds with the graph in Fig. 1. Is the colouring you obtain optimal? (Justify your answer.) **[8 Marks]**

(c) What is an independent set in a graph and what is a maximum independent set? **[3 Marks]**

(d) Suppose that you had an algorithm that always found a maximum independent set in a graph. Explain how you might use this algorithm to obtain a graph colouring algorithm. **[5 Marks]**

(e) Is your algorithm in (d) optimal? (Justify your answer.) **[4 Marks]**

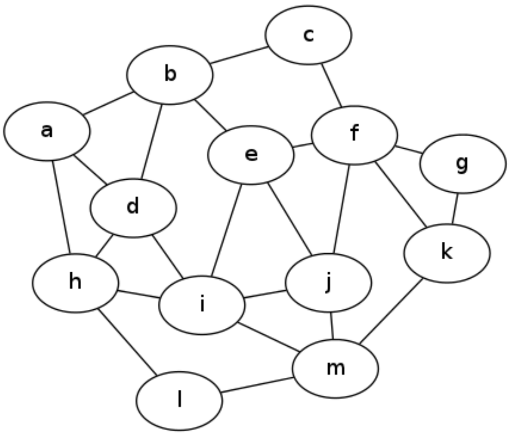

Figure 1: A graph

Question 3

(a) Describe the algorithm Quick-sort using natural language. (Be sure to clearly explain how the list of input numbers is partitioned around the pivot element.) **[14 Marks]**

(b) Describe how partitioning works on the input list of numbers 7, 8, 4, 3, 8, 9. **[6 Marks]**

(c) Does Quick-sort perform efficiently on the list of numbers 1, 2, 3, 4, 5, 6, 7, 8, 9? (Justify your answer.) **[5 Marks]**

End of Section A

continued

Section B Computational Thinking (Prof. Iain A. Stewart)

Question 4

In this question, by 'problem' we mean 'decision problem'.

(a) What does Turing-complete mean as regards a model of computation, and what is the Church-Turing Thesis? What is an unsolvable problem? **[5 Marks]**

(b) What is a non-deterministic algorithm? How does a non-deterministic algorithm accept some input? How do we measure the time taken by some non-deterministic algorithm on some input? **[8 Marks]**

(c) What do we mean by an efficiently solvable problem and an efficiently checkable problem? Is every efficiently checkable problem also efficiently solvable? **[3 Marks]**

(d) Let X and Y be problems. Define what we mean when we say that there is a polynomial-time reduction from X to Y. **[4 Marks]**

(e) An **NP**-complete problem X is a problem that is in **NP** and has the property that: for any other problem Z that is in **NP**, there is a polynomial-time reduction from Z to X. Prove that if Y is in **NP**, X is **NP**-complete, and there is a polynomial-time reduction from X to Y, then Y is also **NP**-complete. **[5 Marks]**

continued

Question 5

(a) Give two different ways in which a software bug can arise and give two different effects of a software bug. **[2 Marks]**

(b) When it comes to establishing program correctness, briefly compare and contrast the different approaches taken in software testing and in formal methods. **[4 Marks]**

(c) In order to execute a high-level program we must first convert it into machine code so that the CPU can 'understand' it. Briefly explain the difference between compilation and interpretation. **[4 Marks]**

(d) Regular expressions are used in the compilation process. Define a regular expression. **[6 Marks]**

(e) Regular expressions define the sets of strings called the regular languages. The regular languages can also be generated using regular grammars. Give another method of defining the regular languages. (There is no need to define any concepts you mention.) **[1 Mark]**

(f) Fix the alphabet $\Sigma = \{a, b, c\}$. Let us say that a string over Σ is a 3-*string* if it starts with an a and ends with a c and:

- every a is followed by a b
- every b is followed by a c
- every c is followed by an a, unless the c terminates the string.

Give a regular expression that defines the set of 3-strings and a regular grammar that generates the set of 3-strings. **[8 Marks]**

END OF PAPER