2910205 Databases

Examiner's report: Zones A and B

Part A

Introduction

There was only one paper to cover all zones.

The CIS205 examination is now available only to candidates retaking after an earlier unsuccessful attempt. Thus the general format of the examination, including the structure and range of questions, is familiar to all candidates and there should be no surprises for them. However, as candidates have experienced difficulty with the assessment in the past, it is not expected that they will have found this year's paper particularly easy, and this was born out by there being no very high overall marks among the small number of candidates. At least some part of each question seemed to give significant problems for a proportion of candidates, with problems ranging from lack of clarity of expression (to the point where intended meaning was not determinable) to simple but significant errors. The level of detail in this report on each question part mostly reflects the degree of difficulty experienced by candidates.

Question 1

a. This was a simple bookwork question and required only a brief statement in relation to use only of internal memory versus the need for use of intermediate external files.

b. A good answer would include mention of Quicksort being O(n.logn) on average and there being no sorting algorithm with better (e.g. O(n), O(n.loglogn), etc) general performance. Thus Quicksort is 'best' in that sense.

c. A detailed specification is expected, both from the phrasing of the question and the number of marks allocated.

   The details of the algorithm, whether in pseudocode or otherwise, should be exact. An example of a significant fault is if gaps are left in algorithms as to when to read in the next item following freezing of items. Also, what to do when all items are frozen should be specified. Any omissions such as these result in many marks lost. The answer should also detail what to do after creating partitions, and several marks are lost if this is not done.

d. This is straightforward bookwork from the designated text or from any one of many others on sorting. A good answer need not give full details of each method, but does need to accurately and completely bring out the relevant points required by the question.

e. Again this is a simple bookwork question, simply answered in terms of the content of standard references. A mention of the underlying motivation for the structure, in terms of handling dynamic files, would

show relevant understanding and would help ensure more credit if there were any slips in completeness of answer or accuracy of expression at the technical level. Relevant points to include are:

- Index records contain pointers to records in external storage where they can be access directly. Hence, no need for records to be physically located in order.
- If the records themselves were sorted then serial access to the records in the file could be achieved without use of the index.
- Sorting the index records would achieve little except possible faster searching of an index block in internal memory. However, savings are likely to be minimal.

Any appropriate answers which consider both the file and the index blocks should be awarded full marks.

Question 2

a. This is just standard bookwork. Relevant points to include are that hashing is a method of calculating the physical location of a record from the value in a key field.

b. There is no single right answer here and anything that is coherent and feasible is appropriate. For example, one approach is to consider an EIN to be a base 36 number (no upper/lower case distinction) by appropriate coding of each symbol. Then any method is acceptable as long as the first stage, the conversion of the identity number to an integer, is reversible. Marks will be lost otherwise.

   Considerations will include looking out for identity numbers which contain, for example, the year of employment. That is, a large number of identity numbers have a common set of numbers or letters. This one consideration is sufficient as long as there is an explanation as to why it could cause problems. Any other reasonable considerations, such as symbols which are never used, may also be awarded marks as long as there is a reasonable explanation as to why they need to be allowed for.

c. This is essentially bookwork, covered in any standard text on the subject. In any good answer, there must be mention of volatility. Failure to take account of volatility, which may not necessarily be explicitly named, would lose many or most of the marks. The explanation should then discuss overflow and the increased likelihood of overflow and consequent slowing of retrieval.

   For example, a file with 90 peer cent packing might operate satisfactorily. If the file is not added to frequently, then it is likely to remain satisfactory for some time. If the operation of the file becomes unsatisfactory (i.e. retrieval times are unacceptable) then rehash using more storage space to achieve a lower packing density.

d. This is straightforward bookwork from any standard source. The distinction between primary and secondary clustering must be clear via the descriptions given.

e. Again, this is standard bookwork. All that is required is clarity of description and an accurate statement of primary/secondary clustering susceptibility.

Question 3

a. Provided that the nature of the medium is understood then all that is required is the application of simple techniques. Example points to make include:

   - Hashing – very fast retrieval on one field.
   - Static indexes – several indexes can be set up so as to retrieve records using several different fields. Not as fast as hashing.

   By contrast, B-trees are a dynamic index structure and irrelevant to a CD-ROM which cannot be written to. Thus any case for using a B-tree would lose marks.

   Again, as the medium is read only, any mention of packing density or volatility as a reason for using or not using hashing would lose marks.

b. The question says very precisely what must be defined and so marks will be lost for any omissions. The important aspect of this question is to see an understanding of what is stored on external storage, what is copied into internal memory and how it is used. Any answer which shows theoretical diagrams with pointers and lines but fails to relate these pointers to actual physical locations and movement of records from external to internal memory would be awarded at most about half of the marks.

c. This is straightforward bookwork, with the required example enabling candidates to show understanding and to earn good marks if their written description of the general structure is unclear, though clarity there too is needed for full marks.

   For example, in relation to the last part, for 'self-balancing', an indication that all nodes of a B-tree have an upper and lower limit and that all leaves are the same distance from the root, should be evident. Although there is no need for a description of deletion and concatenation, there should at least be a mention that there is an inverse process to insertion, which also maintains the balanced feature of the B-tree.

d. This is standard bookwork and so the answer can reflect any standard source that has been read by the candidate.

Question 4

a. There is no one right answer for the question. What is looked for is coherence in choice and a sound case made for each choice or decision. Example matters that might be included, or choices made, are:

   - Hash on the catalogue number and create threads for the other features.
   - The answer will need to give a record structure and show a clear understanding of what a threaded file is.

   Any answer which considers each aspect separately and does not take into account that there is just one file loses nearly a third of the marks immediately.

b. For example, maintaining lengths of threads, or even mention of fully inverting the file, gain full marks as long it is shown how why they would be useful and is appropriate in relation to the answer for (a).

c. This part must simply show an awareness of what is required in inserting and deleting records in a threaded file and the amendments to the threads (or whatever appropriate in relation to (a)). Possible mention of next and previous record fields could appear but is not vital.

Part B

Question 6

Question 6 a) asked candidates to list four characteristics of procedural languages and for each of these characteristics, give an example to show how they occur in a procedural language of their choice.

This is mainly bookwork with candidates expected to know that procedural languages are tied with low level computer operation and to be able to give examples of a program as a description of these operations, with operations on data objects and often using assignment of values to variables.

Part b) gives candidates the opportunity to show their understanding of typing, scoping and the different types of procedure call. Each of these is an important area for candidates to understand before they can be expected to produce bug free code in a procedural language.

Question 7

Question 7 a) moves our focus onto declarative languages and in particular to Prolog. Prolog is problem oriented; it separates data from execution, works at a higher level than procedural languages and has declarative and procedural readings.

These characteristics of Prolog are important for candidates to comprehend.

Part b) gives a Prolog implementation of append and invites candidates to trace two given queries. Tracing is an essential skill in all programming but is rather more complex for Prolog. Having facility with Prolog tracing is necessary and is a sign of a good understanding of a program's procedural reading.

Part c) asks candidates to write a simple Prolog program which reverses the order of elements in a list.

Candidates were given 'append' in part b) and should have had little difficulty producing the required two clauses to complete the task.

Question 8

For Question 8 a), among the advantages of Prolog over procedural languages given in the course guide are: more readable; allows focus on specification of answer; easier to debug in some cases; more compact; own specification language. Candidates are able to give others if they wish. On the other hand some imperative languages, such as Pascal or C, may be better than declarative ones such as Prolog in terms of: faster code; ease of debugging in some cases; and allowing type checking.

The concept and process of unification is one of the most important in Prolog and one cannot claim to know the language until unification (matching) is mastered. Good candidates will know that: i) a constant matches itself; ii) a variable matches anything; iii) two structures match if they have the same functor and arity and their arguments match under some fixed substitution of variables.

Part d) required candidates to match pairs of terms, give the instantiations required for them to match or, if they cannot be made to match, an explanation of why they do not match. This requires just an application of the matching rules noted as answer to part c) of the question.

Lists and their representation in both 'dot notation' and as trees is the topic of part d).

It is the ability of lists to represent arbitrary trees that give them such a prominent place in Prolog programming.

Question 9

In Question 9 a) a description of the meaning of the terms polymorphism and overloading in the context of Standard ML is required. These topics provide an area where important differences arise between the languages of the subject: Java, Prolog and Standard ML.

Part b) tests the candidate's knowledge of the Standard ML terms **andalso**, **else** and **orelse**. Examples of their use are required for full answers.

In Standard ML, execution is thought of as a process of reduction of terms (built from constructors) to normal form, and part c) requires a description of the constructors for lists and a description of their use. For reduction to normal form to be useful this form must be unique for each term.

Candidates should have a great deal of experience in defining functions in Standard ML and should thus know the syntax of such a definition. Using list reversal as an example, they are to explain this syntax.

Question 10

The reduction process is looked at in more detail in Question 10 where part a) uses the evaluation of the expression 1+2 as an example, requiring the candidate to describe the process of rewriting to produce a normal form. Candidates should have no difficulty giving the meaning of the terms **normal form**, **redex** and **reduction rules** though there is often some confusion amongst weaker candidates about the distinction between a rewrite rule and a reduction rule.

In part b) candidates show their understanding of this process by giving a step by step evaluation of: *if (3*2) = (2 * 3)* **then "y" else "n"** to its normal form "*y*": **string**

The topic of 'user defined types' and how they are defined in standard ML is the subject of part c). Candidates are to use a tree of integers as an example and to give examples of two such trees.

The final part, part d), uses this representation of trees and asks for a function 'member' that takes a tree and an integer and returns true if and only if that integer occurs in the tree.