COMPUTER SCIENCE TRIPOS Part II (General)
DIPLOMA IN COMPUTER SCIENCE

Monday 2 June 2008 1.30 to 4.30

PAPER 10 (PAPER 1 OF DIPLOMA IN COMPUTER SCIENCE)

*Answer **five** questions.*

*Submit the answers in five **separate** bundles, each with its own cover sheet. On each cover sheet, write the numbers of **all** attempted questions, and circle the number of the question attached.*

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

STATIONERY REQUIREMENTS
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
None

1 Foundations of Programming

(*a*) Briefly describe *ASCII* and *Unicode* and draw attention to any relationship between them. [3 marks]

(*b*) Briefly explain what a *Reader* is in the context of reading characters from data. [3 marks]

A novice programmer has written the following copying program which is intended simply to read characters from data one at a time and to write them out.

```
public class Copying
 { public static void main(String[] args)
    { int ch;
      while ((ch = System.in.read()) != -1)
       { System.out.printf("%c", (char) ch);
       }
    }
 }
```

(*c*) Unfortunately this program causes a compile-time error. Explain what the problem is and modify the code so that the program compiles. [3 marks]

(*d*) The amended program correctly copies data in simple cases but is found to fail when exotic characters are encountered. Why is this? [3 marks]

(*e*) By exploiting a Reader, rewrite the program so that exotic characters no longer cause any problems. [8 marks]

2 Foundations of Programming

(*a*) Outline the use of `enum` in Java programs and explain any similarities to and any differences from a Java class. [6 marks]

The following fragment of code shows an early attempt to write and test a Java program to simulate the paper-scissors-stone game. In the method `main()` each of the identifiers `a` and `b` is assigned one of the items, paper, scissors or stone and the `printf()` method then says whether the first beats, draws with or loses against the second. [The rule is that paper beats stone, scissors beat paper and stone beats scissors.]

```
public class Game
 { public static void main(String[] args)
    { Item a = allocateItem();
      Item b = allocateItem();
      int res = a.versus(b);
      System.out.printf("%s %s %s%n", a, res>0 ? "beats" :
                                    res==0 ? "draws with" :
                                          "loses against", b);
    }

   private static Item allocateItem()
    { ...
    }
 }

enum Item
 { PAPER...
```

The three possibilities `PAPER`, `SCISSORS` and `STONE` are declared in `enum Item` along with a constructor, the method `versus()` and any necessary additional data fields.

(*b*) Using the method `Math.random()` or otherwise, provide a body for the method `allocateItem()` such that the method returns an `Item` `PAPER`, `SCISSORS` or `STONE` equiprobably. [6 marks]

(*c*) Complete the `enum Item` so that the statements in the method `main()` operate as intended. [8 marks]

3 Floating-Point Computation

(*a*) Write a function in a programming language of your choice that takes a (32-bit IEEE format) `float` and returns a `float` with the property that: given zero, infinity or a positive normalised floating-point number then its result is the smallest normalised floating-point number (or infinity if this is not possible) greater than its argument. You may assume functions `f2irep` and `irep2f` which map between a `float` and the same bit pattern held in a 32-bit integer. [6 marks]

(*b*) Briefly explain how this routine can be extended also to deal with negative floating-point values, remembering that the result should always be greater than the argument. [2 marks]

(*c*) Define the notions of *rounding error* and *truncation error* of a floating-point computation involving a parameter h that mathematically should tend to zero. [2 marks]

(*d*) Given a function f implementing a differentiable function that takes a floating-point argument and gives a floating-point result, a programmer implements a function

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

to compute its derivative. Using a Taylor expansion or otherwise, estimate how rounding and truncation errors depend on h. You may assume that all mathematical derivatives of f are within an order of magnitude of 1.0. [8 marks]

(*e*) Suggest a good value for h given a double-precision floating-point format that represents approximately 15 significant decimal figures. [2 marks]

4 Programming in C and C++

A hardware engineer stores a FIFO queue of bits in an `int` on a platform with 32-bit `int`s and 8-bit `char`s using the following C++ class:

```
class BitQueue {
  int valid_bits; //the number of valid bits held in queue
  int queue;      //least significant bit is most recent bit added
 public:
  BitQueue(): valid_bits(0),queue(0) {}
  void push(int val, int bsize);
  int pop(int bsize);
  int size();
};
```

(*a*) Write an implementation of `BitQueue::size`, which should return the number of bits currently held in `queue`. [1 mark]

(*b*) Write an implementation of `BitQueue::push`, which places the `bsize` least significant bits from `val` onto `queue` and updates `valid_bits`. An exception should be thrown in cases where data would otherwise be lost. [5 marks]

(*c*) Write an implementation of `BitQueue::pop`, which takes `bsize` bits from `queue`, provides them as the `bsize` least significant bits in the return value, and updates `valid_bits`. An exception should be thrown when any requested data is unavailable. [4 marks]

(*d*) The hardware engineer has built a communication device together with a C++ library function `send` to transmit data with the following declaration:

```
void send(char);
```

Use the `BitQueue` class to write a C++ definition for:

```
void sendmsg(const char* msg);
```

Each of the characters in `msg` should be encoded, in index order, using the following binary codes: `'a'`=`0`, `'b'`=`10`, `'c'`=`1100`, and `'d'`=`1101`. All other characters should be ignored. Successive binary codes should be bit-packed together and the code `111` should be used to denote the end of the message. Chunks of 8-bits should be sent using the `send` function and any remaining bits at the end of a message should be padded with zeros. For example, executing `sendmsg("abcd")` should call the `send` function twice, with the binary values `01011001` followed by `10111100`. [10 marks]

5 Computer Graphics and Image Processing

(*a*) Most liquid crystal displays divide a pixel into three sub-pixels coloured red, green, and blue. Explain why this is so. [4 marks]

(*b*) Some liquid crystal displays divide a pixel into four sub-pixels coloured red, green, blue, and white. Explain why this might be useful, what advantages it has, and what limitations it has. [6 marks]

(*c*) Compare and contrast half-toning and error diffusion. Include in your answer an explanation of the situations in which each is superior to the other. [6 marks]

(*d*) One method of anti-aliasing is to sample at high resolution, $n \times n$ higher than the final image, and then to average each block of $n \times n$ pixels to give a single pixel value. Discuss the advantages and disadvantages of using

(*i*) Gaussian blurring, and

(*ii*) median filtering

in place of simple averaging. [4 marks]

6 Introduction to Functional Programming

(*a*) Write an SML function

```
propercuts:  (α list) → (α list * α list) list
```

that given a list ℓ outputs the list of all pairs of non-empty lists (ℓ_1, ℓ_2) such that $\ell_1 \texttt{@} \ell_2 = \ell$. [5 marks]

(*b*) Consider the following datatypes

```
datatype α tree = leaf of α | node of α tree * α tree ;

datatype α symbol = Lbracket | Rbracket | token of α ;
```

and the SML function

```
rep:  α tree → (α symbol) list
```

that represents a binary tree as a list of symbols, according to the following definition:

```
fun rep ( leaf x ) = [ token x ]
  | rep ( node(l,r) )
    = [ Lbracket ] @ (rep l) @ (rep r) @ [ Rbracket ] ;
```

Write an SML function

```
istree:  (α symbol) list → bool
```

that given a list of symbols ℓ outputs `true` if there exists a (necessarily unique) tree t such that `rep`$(t) = \ell$, and outputs `false` otherwise. [10 marks]

(*c*) Define the SML functions

```
infix @ ;
@: α list * α list → α list ;

map: (α → β) → α list → β list ;
```

and rigorously argue for the correctness of the following identity:

```
map f (ℓ1 @ ℓ2) = (map f ℓ1) @ (map f ℓ2)   : β list
```

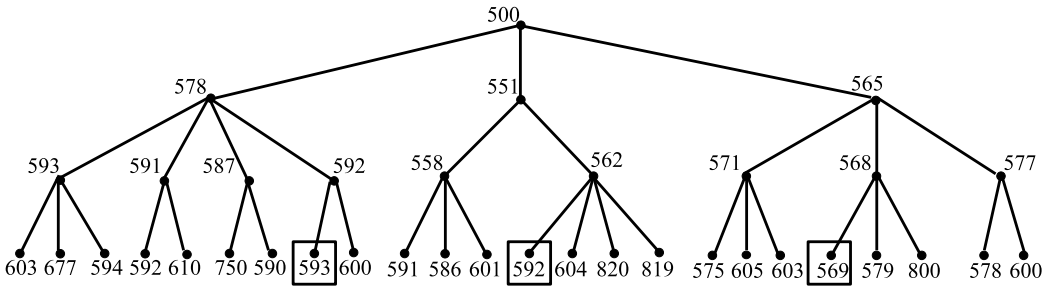
for all $f : \alpha \to \beta$ and $\ell_1, \ell_2 : \alpha$ `list`.

[5 marks]

7 Artificial Intelligence I

(*a*) Give a general description of the operation of the *Recursive Best-First Search* (*RBFS*) algorithm. [6 marks]

(*b*) Consider the following search tree.



The numbers by the nodes denote the sum of some path cost and heuristic. The boxed nodes are goals. Describe in detail the way in which the RBFS algorithm searches this tree. Your answer should indicate the order in which nodes are expanded, the reason that this order is used, and should state which of the three goals is found and why. Note that smaller numbers represent more desirable nodes. [12 marks]

(*c*) What shortcoming of the $A^\star$ algorithm does the RBFS algorithm address, and how does it achieve this? [2 marks]

8 Introduction to Security

(*a*) A source of secure, unpredictable random numbers is needed to choose cryptographic keys and nonces.

(*i*) Name *six* sources of entropy that can be found in typical desktop-computer hardware to seed secure random-number generators. [4 marks]

(*ii*) What sources of entropy can a smartcard chip, like the one in your University Card, access for this purpose? [4 marks]

(*b*) As Her Majesty's prime hacker "001", on a mission deep inside an enemy installation, you have gained brief temporary access to a secret chip, which contains a hardware implementation of the DES encryption algorithm, along with a single secret key. You connect the chip to your bullet-proof laptop and quickly manage to encrypt a few thousand 64-bit plaintext blocks of your choice, and record the resulting 64-bit ciphertext blocks. You are unable to directly read out the DES key K used in the chip to perform these encryptions and you will not be able to leave the site without knowing K. But you know that all S-boxes in the last DES round are supplied in this chip via a *separate* power-supply pin. When you create a short-circuit on that pin, the encryption progresses as normal, except that the output of all S-boxes in the last round changes to zero.

(*i*) Explain briefly the role of an S-box and the structure of a single round in DES. [4 marks]

(*ii*) How can you find K, considering that your available time and computing power will not permit you to search through more than 10^9 possible keys? [8 marks]

9 Data Structures and Algorithms

(*a*) Take an initially empty hash table with five slots, with hash function $h(x) = x \bmod 5$, and with collisions resolved by chaining. Draw a sketch of what happens when inserting the following sequence of keys into it: `35, 2, 18, 6, 3, 10, 8, 5`.
[You are not requested to draw the intermediate stages as separate figures, nor to show all the fields of each entry in detail.] [3 marks]

(*b*) Repeat part (*a*) but with the following three changes: the hash table now has ten slots, the hash function is $h(x) = x \bmod 10$, and collisions are resolved by linear probing. [3 marks]

(*c*) Imagine a hash table implementation where collisions are resolved by chaining but all the data stays within the slots of the original table. All entries not containing key–value pairs are marked with a Boolean flag and linked together into a free list.

(*i*) Give clear explanations on how to implement the `set(key, value)` method in expected constant time, highlighting notable points and using high-level pseudocode where appropriate. Make use of doubly-linked lists if necessary. [8 marks]

(*ii*) Assume the hash table has 5 slots, is initially empty and uses the hash function $h(x) = x \bmod 5$. Draw five diagrams of the hash table representing the initially empty state and then the table after the insertion of each of the following key–value pairs: `(2, A), (2, C), (12, T), (5, Z)`. In the final diagram, draw all the fields and pointers of all the entries. [6 marks]

10 Operating System Foundations

(*a*) Assume a 32-bit architecture with hardware support for paging, in the form of a translation lookaside buffer (TLB), but no hardware support for segmentation. Assume that the TLB is shared rather than flushed on process switching and that the operating system designers are supporting "soft" segments.

(*i*) In addition to page number and page base, what fields would you expect to find in each TLB register?
How would each of these be used? [4 marks]

(*ii*) What fields would you expect to find in a process page table?
How would each of these fields be used? [6 marks]

(*b*) (*i*) Outline the function of a timing device. [2 marks]

(*ii*) Why are timers essential in multiprogramming operating systems? [2 marks]

(*iii*) Explain the operation of a multi-level feedback queue in process scheduling. [6 marks]

END OF PAPER