

CAMBRIDGE INTERNATIONAL EXAMINATIONS
General Certificate of Education
Advanced Subsidiary Level and Advanced Level

COMPUTING

9691/02

Paper 2: Practical Tasks

May/June 2004

READ THESE INSTRUCTIONS FIRST

Write your Centre number, candidate number and name on all the work you hand in.
Write in dark blue or black pen on both sides of the paper.
Do not use staples, paper clips, highlighters, glue or correction fluid.

Answer **all** questions.

The number of marks is given in brackets [] at the end of each question or part question.

This document consists of **5** printed pages and **3** blank pages.



Answer **all** questions.

Task 1 [29 marks]

This is a software development task and an implementation task.

An English department in a college expects students to read a number of books which the Department keeps in its library. The Department wishes to keep details of books borrowed by students to ensure that the students read the books set by the tutors. A student borrows many books and a book is borrowed by many students.

A database consisting of the tables Student, Book and StudentBook is to be created. The StudentBook table is used to link the other two tables and has the following attributes (fields), each with the specified data type.

Attribute (field)	Data Type
StudentID	Numeric type
BookID	Numeric type
Date Borrowed	Date type

The key is a compound key formed from StudentID and BookID.

(a) Create a table called Student to hold details of all the students.

(i) Describe the

- purpose of each attribute,
- data type of each attribute.

[6]

(ii) Identify the key.

[1]

(b) Create a table called Book to hold the details of the books.

(i) Describe the

- purpose of each attribute,
- data type of each attribute.

[8]

(ii) Identify the key.

[1]

(c) Create data for the three tables. You should have at least 20 students and 15 books. The data should be suitable for testing the tables. Include a print-out of your tables in your documentation.

[7]

(d) Create an interface that allows a user to input a book's ID and prints a report showing a list of the students who have borrowed that book and the dates on which each student borrowed the book. Provide hard copy evidence that the report is produced.

[3]

(e) Create an interface that allows a user to input a student's ID and prints a report showing a list of the books the student has borrowed and the dates on which each book was borrowed. Provide hard copy evidence that the report is produced.

[3]

Task 2 [10 Marks]

This is an algorithm trace task. No implementation is required.

L, R and Data are one-dimensional arrays, each with 5 elements numbered from 1 to 5. The contents of the three arrays are shown in Fig. 2.1.

Index	L	Index	R	Index	Data
1	2	1	3	1	DOG
2	0	2	4	2	BEAR
3	5	3	0	3	HORSE
4	0	4	0	4	CAT
5	0	5	0	5	GIRAFFE

Fig. 2.1

In these arrays, L(3) contains the number 5, R(2) contains the number 4 and Data(1) contains the data DOG.

Read the following algorithm.

```

T = 0
P = 1
REPEAT
  WHILE P <> 0 DO
    T = T + 1
    S(T) = P
    P = L(P)
  END WHILE
  IF T > 0 THEN
    Finished = FALSE
    P = S(T)
    T = T - 1
    OUTPUT Data(P)
    P = R(P)
  ELSE
    Finished = TRUE
  END IF
UNTIL Finished
EXIT

```

Task 2 is continued on page 4 overleaf.

Complete the trace table shown in Fig. 2.2. It is only necessary to enter values in the table when they change. You should work from left to right and top to bottom. The first six lines are completed for you.

T	P	S(1)	S(2)	Finished	Output
0	1				
1		1			
	2				
2			2		
	0			FALSE	
	2				

Fig 2.2

[10]

Task 3 [21 Marks]

This is a software development task and an implementation task.

Roman numerals are still used for certain limited purposes such as copyright dates. The symbols used are I =1, V = 5, X = 10, L = 50, C = 100, D = 500 and M =1000.

One definition of Roman numbers is

- the letters must be in descending order of value,
- the value of the number is the sum of all the letters,
- each letter must be used a minimum number of times.

This is known as the additive rule and is the only rule to be used in this question.

For example

MCX = 1000 + 100 + 10 = 1110;

MXC is invalid because of wrong order;

IIII = 1 + 1 + 1 + 1 =4;

IIII is invalid because it can be replaced with V.

A manufacturer wishes to make a new calculator that allows a user to input a number using Roman numerals and displays the number in both Roman and Arabic forms.

The calculator must

- allow a user to enter the Roman numerals up to a value of 7000 (MMMMMMM),
- prevent the user entering an invalid numeral,
- display the number entered using Roman numerals,
- update this display as each Roman numeral is entered,
- display the number entered using Arabic numerals,
- update this display as each Roman numeral is entered,
- allow the user to clear both displays, the Roman number and the Arabic number.

The manufacturer wishes to use a simulation of this calculator so that the user interface can be tested.

- (a) Design a user interface that can be used to test the above facilities. Provide hard copy showing this design. (A screen dump is acceptable.) [5]
- (b) Using a high-level language of your choice, implement and test your design to show that all the above facilities work correctly.
- (i) Provide hard copy of your annotated program, stating the language and version used. [5]
- (ii) Provide a table showing your test data and the expected and actual results. [6]
- (iii) Supply hard copy evidence of testing your solution. (Screen dumps are acceptable, but must show the data entered and the results.) [5]

