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

COMPUTING **9691/02**

Paper 2: Practical Tasks

October/November 2005

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 **6** printed pages and **2** blank pages.

IB05 11_9691_02/4RP

Answer **all** questions.

Task 1 [22 marks]

This is a design and implementation task.

A library requires a database to hold details of its books and authors. An author may write many books and a book may be written by many authors.

Authors have an address and a telephone number. The telephone number consists of 11 digits such as 01543564329. The author's name, address and telephone number are the only personal data to be stored.

Books have a book number consisting of two letters and four digits such as CS1002. Each book has a title and a publisher.

(a) Design a database, consisting of three tables, to store the data. You should include evidence showing the attributes of the tables together with their data types and any validation rules you have used. Screen dumps are acceptable for this purpose. [11]

Create sufficient data to complete the rest of this task. Provide copies of your completed tables. Screen dumps are acceptable for this purpose.

(b) **(i)** Create a form that will allow the user to add an author to the appropriate table.

(ii) Create a form that will allow the user to add a book to the appropriate table.

(iii) Create a form that will allow a user to link an author to a book. The form should allow the user to choose an author from a list and to choose a book from a list. [6]

(c) Create a query that, when run, requests a user to select an author and then lists all the books written by that author. Include evidence that your query works correctly. [2]

(d) Create a report that lists the authors for each book. The report should group the information on book number and have appropriate headings. Include evidence that your report is correct. [3]

Task 2 [18 Marks]

This is a white box test task. No implementation is required.

Read the following algorithm for a procedure, called Validate, in which each step has been numbere

```
Validate(aString)
1  Length = Len(aString)
2     IF Length = 0 THEN
3        Output "Empty string is not allowed"
4     ELSE
5        OK = TRUE
6        DP = FALSE
7        Count = 1
8        Ch = 1st character of aString
9        IF Ch < "i" OR Ch > "n" THEN
10          OK = FALSE
11       ELSE
12          WHILE Count < Length AND OK DO
13             Count = Count + 1
14             Ch = next character in aString
15             IF Ch = "." THEN
16                IF DP THEN
17                   OK = FALSE
18                ELSE
19                   DP = TRUE
20                ENDIF
21             ELSE
22                IF Ch < "a" OR Ch > "z" THEN
23                   OK = FALSE
24                ENDIF
25             ENDIF
26          ENDWHILE
27       ENDIF
28       IF OK THEN
29          Output "Valid string"
30       ELSE
31          Output "Invalid string"
32       ENDIF
33    ENDIF
34 END PROCEDURE Validate
```

The function Len(aString) returns the number of characters in aString. For example, if aString = "Computer", Len(aString) = 8.

When the procedure is called with

Validate("kg")

the lines

1, 2, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 21, 22, 24, 25, 26, 12, 26, 27, 28, 29, 30, 32, 33, 34

are executed and the output is

Valid string

The example test above can be described as shown in Table 2.1.

Path	**Test Data**	**Expected Output**
1, 2, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 21, 22, 24, 25, 26, 12, 26, 27, 28, 29, 30, 32, 33, 34	kg	Valid string

Table 2.1

Using a table similar to Table 2.1, write down the lines that are executed and the output when the procedure is called with

(i) Validate(".doc"); [4]

(ii) Validate("m.d"); [7]

(iii) Validate("j.b.w"). [7]

Task 3 [20 marks]

This is a design and implementation task.

You are to design and create a very simple octal calculator using a high-level language of your choice. An octal calculator can only use the digits 0 to 7. This calculator will allow the user to enter two unsigned octal numbers and will then allow the user to add, subtract, multiply or divide these numbers. In the case of division, the result should be truncated to a whole number. This is equivalent to integer division. The result is to be displayed in octal.

To do this you are advised to follow these steps.

(a) Design an interface that will allow the user to input two unsigned octal numbers and will display them in two boxes. The user should then be able to choose addition, subtraction, multiplication or division. There should be a box to show the result of the calculation. The user should also be able to clear all the boxes. (No processing is required at this stage.) [4]

(b) Create a function, called OctalToDecimal, that will accept an octal number as a parameter and will return its decimal equivalent. For example,

OctalToDecimal(253) will return the decimal value 171.

You may use the following algorithm to do this.

```
Input the octal number as a string
Total = 0
For each octal digit in the string, starting on the left,
    Total = Total * 8 + value of next digit
Return Total
```

You should clearly annotate your code and use meaningful names for variables and other objects such as buttons and text boxes (if you use them). You should include a copy of your code as evidence. [4]

(c) Create a function, called DecimalToOctal, that will accept a decimal number as a parameter and will return its octal equivalent. For example,

DecimalToOctal (171) will return the octal value 253.

You may use the following algorithm to do this.

```
Input the decimal number called Number
Octal = ""                  'Empty string
Repeat
    Remainder = remainder after Number is divided by 8
    Number = Whole part of Number divided by 8
    Octal = String value of Remainder & Octal
Until Number = 0
```

& means concatenation. For example

"5" & "263" = "5263"

You should clearly annotate your code and use meaningful names for variables and other objects such as buttons (if you use them). You should include a copy of your code as evidence. [4]

(d) Create code for each of the operations of addition, subtraction, multiplication and division. You are advised to change the octal numbers input by the user to decimal, do the calculation and then convert the result back to octal before displaying the result. You should include a copy of your code as evidence.

(e) Create a set of test data that shows that the operations of addition, subtraction, multiplication and division work and provide evidence that you have used this data. Screen dumps are acceptable but must show the data entered and the result of the operation. [5]

BLANK PAGE

BLANK PAGE