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

COMPUTING   **9691/02**

Paper 2 Practical Programming Techniques   **For Examination from 2011**
SPECIMEN PAPER

2 hours

Additional Materials: Answer Booklet/Paper

READ THESE INSTRUCTIONS FIRST

If you have been given an Answer Booklet, follow the instructions on the front cover of the Booklet.
Write your Centre number, candidate number and name on all the work you hand in.
Write in dark blue or black pen.
You may use a soft pencil for any diagrams, graphs or rough working.
Do not use staples, paper clips, highlighters, glue or correction fluid.

Answer **all** questions.

At the end of the examination, fasten all your work securely together.
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 **1** blank page.

[Turn over

1 A database of properties in an estate agency displays, on the screen, a form to allow the user to enter queries. The following details are displayed on the screen:

- a company logo
- a space for the name of the customer
- 2 questions about the properties interested in with space for the answers
- a means of going on to the next screen and also a means of returning to the previous one

(a) Draw a suitable layout for the screen. [6]

The system stores details of the last 100 queries made. These are held in a file which contains, for each query made, the following:

- the name of the customer
- the number of bedrooms requested
- the date on which the query was made
- whether the customer asked for details of a property or not

(b) Copy and complete the following table.

	Data type	Size of field (bytes)
Name		
Number of bedrooms		
Date		
Details requested?		

[8]

(c) Estimate the size, in KB, of the file of 100 records. Show your working. [5]

(d) The records are held in a serial file.

Describe the process for adding a new query into the file that already contains 100 queries. There should be no more than 1 entry per customer. If there is already a record for that customer it should be replaced by the new one. [5]

2 A program has been written which inputs the marks gained by 1000 candidates in an examination. The top mark possible is 100. The program calculates the mean mark and outputs the highest mark, the lowest mark and the mean.

(a) It is decided to test the program by using sets of test data containing four marks each time.

Explain why the testing was carried out using only four marks at a time. [2]

(b) Using a table like the one below, give **three** separate test cases for testing the program.

	Input data	Reason for test	Expected result
Test 1			
Test 2			
Test 3			

[9]

3 A town election is held to elect a new mayor.

The people in the town can vote for whoever they prefer from the three candidates A, B, C.

The voting is done by each voter pressing one of a set of buttons labelled A, B and C in the voting booth. This acts as input to a computer program.

An early version of the software assumes that there are 1000 people who will be voting. It uses an array, VOTES() to store the votes that are cast. VOTES() is an array of 1000 characters, A, B or C.

A second array, CANDIDATE_TOTALS(), contains three integers and is used with the array, VOTES() in the following algorithm:

```
01 FOR i = 1 TO 1000
02   IF VOTES(i) = "A" THEN
03       CANDIDATE_TOTALS (1) = CANDIDATE_TOTALS (1) + 1
04       ELSE
05       IF VOTES(i) = "B" THEN
06           CANDIDATE_TOTALS (2) = CANDIDATE_TOTALS (2) + 1
07           ELSE
08           CANDIDATE_TOTALS (3) = CANDIDATE_TOTALS (3) + 1
09       ENDIF
10   ENDIF
11 NEXT i
12 OUTPUT A,B,C
```

(a) (i) Explain why it will be necessary to initialise the array CANDIDATE_TOTALS () before the algorithm is run. [1]

(ii) Write a FOR loop which can be used to initialise the array CANDIDATE_TOTALS() at the start of the algorithm. [2]

(iii) Explain what happens when the program based on this algorithm is executed. [4]

(iv) By stating the type of operator in each case, explain why the use of the '=' signs in lines 2 and 3 are different. [3]

(v) Line 12 is meant to output the total votes for A, B and C. It does not work.

Rewrite line 12 to produce the correct result. [2]

(b) The following algorithm is written to determine which of A, B and C gets the highest vote.

```
01 IF CANDIDATE_TOTALS (1) > CANDIDATE_TOTALS (2) THEN
02   IF CANDIDATE_TOTALS (1) > CANDIDATE_TOTALS (3) THEN
03        OUTPUT "A"
04        ELSE
05        OUTPUT "C"
06   ENDIF
07   ELSE
08   IF CANDIDATE_TOTALS (2) > CANDIDATE_TOTALS (3)THEN
09        OUTPUT "B"
10        ELSE
11        OUTPUT "C"
12   ENDIF
13 ENDIF
```

   (i) Some people do not vote and the result of a particular vote is that all of A, B and C receive equal votes.

     State the line numbers that will be executed by the algorithm and which of A, B or C will be output. [4]

   (ii) Explain how the algorithm would need to be altered to deal correctly with two or three of the candidates receiving equal votes. Do **not** produce the algorithm. [4]

4 A garage is having software produced to calculate the bills for its customers. A text file contains a list of all the servicing for a customer. An extract of this file is given below.

| Date	Duration	Parts
01/01/2007	01:09	$17.07
03/08/2007	02:52	$29.27
12/02/2007	04:13	$43.15

The garage charges for labour by charging for every half hour, or part of a half hour.

(a) (i) Explain what is meant by a subroutine. [3]

   (ii) Functions and procedures are both examples of subroutines.

     Explain how a function differs from a procedure. [2]

   (iii) The development team decides to divide the problem into modules as shown in the diagram below.
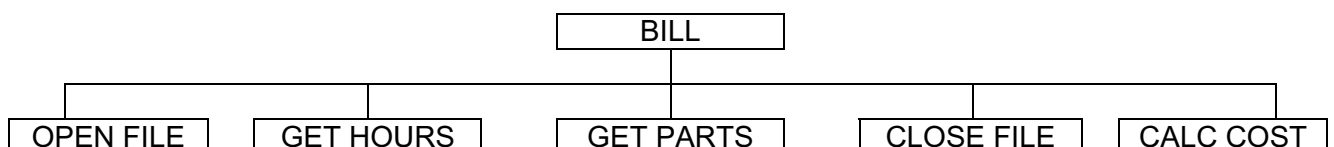
     Draw part of the next level of the diagram by dividing the module 'GET HOURS' into further sub tasks.
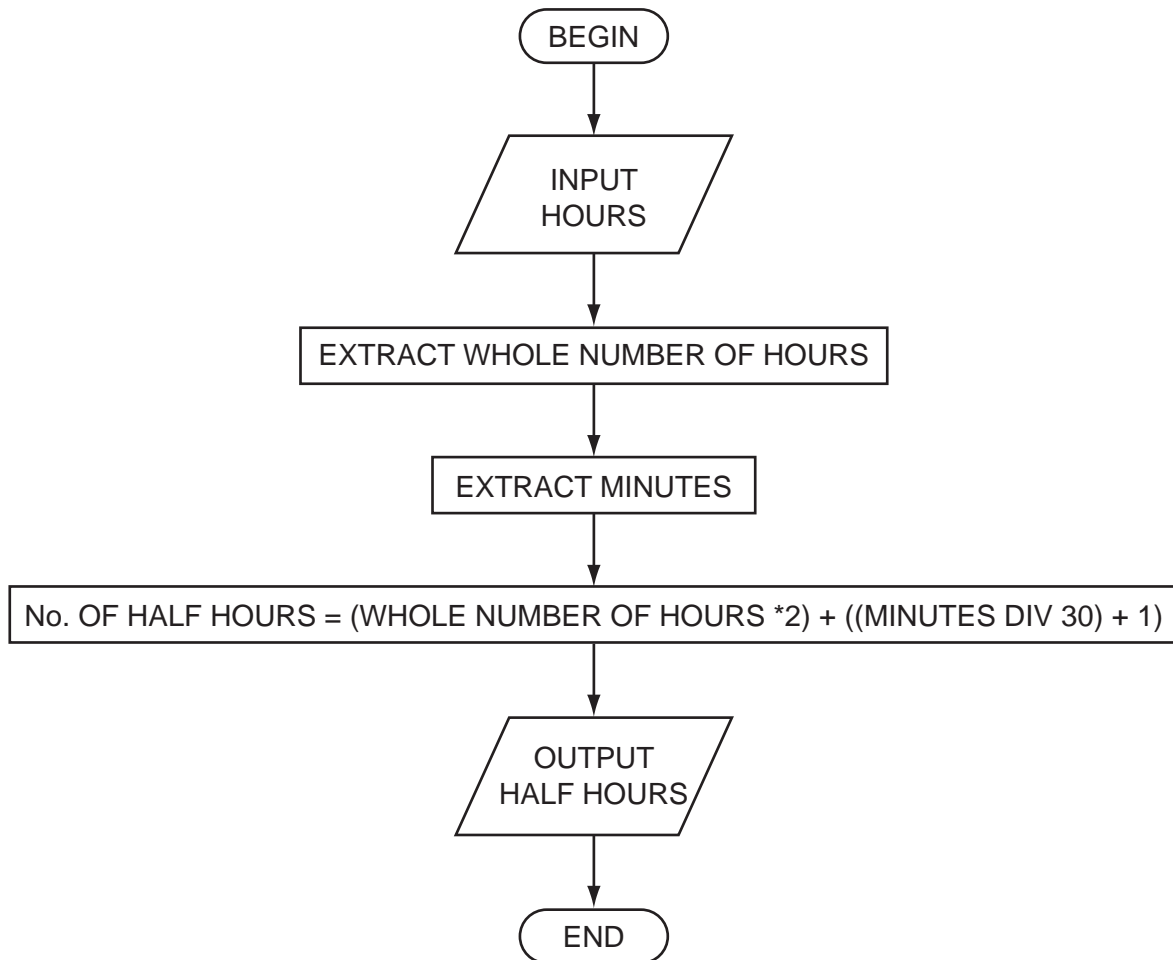
     Do **not** attempt to divide any of the other modules. [4]

(b) (i) Describe **two** programming techniques that can be used when writing a program to help others to understand the code. [4]

(ii) The software for the garage includes a function which takes HOURS as a **string** and returns the number of half hours.

For example, if the input is "1:30" the output will be 3; if the input is 2:52 the output will be 6.

Here is an algorithm for this function.

```
                    ┌──────────┐
                    │  BEGIN   │
                    └──────────┘
                         │
                         ▼
                   ╱─────────────╲
                  ╱    INPUT       ╲
                  ╲    HOURS       ╱
                   ╲─────────────╱
                         │
                         ▼
          ┌──────────────────────────────────────┐
          │  EXTRACT WHOLE NUMBER OF HOURS        │
          └──────────────────────────────────────┘
                         │
                         ▼
              ┌──────────────────────┐
              │   EXTRACT MINUTES     │
              └──────────────────────┘
                         │
                         ▼
┌──────────────────────────────────────────────────────────────────────────┐
│ No. OF HALF HOURS = (WHOLE NUMBER OF HOURS *2) + ((MINUTES DIV 30) + 1)     │
└──────────────────────────────────────────────────────────────────────────┘
                         │
                         ▼
                   ╱─────────────╲
                  ╱   OUTPUT       ╲
                  ╲  HALF HOURS    ╱
                   ╲─────────────╱
                         │
                         ▼
                    ┌──────────┐
                    │   END    │
                    └──────────┘
```

Rewrite this function in a high level language using appropriate string manipulation functions, ensuring that your code can be easily followed by another programmer.

You must state the high level language that you use. [7]

6

BLANK PAGE

www.theallpapers.com